APPLICATION OF MULTIDISCIPLINARY DESIGN OPTIMISATION FRAMEWORKS FOR ENGINE MAPPING AND CALIBRATION

M.R. KIANIFAR

PhD

UNIVERSITY OF BRADFORD

2014

Application of Multidisciplinary Design Optimisation Frameworks for Engine Mapping and Calibration

Mohammed Reza KIANIFAR

Submitted for the Degree of

Doctor of Philosophy

Faculty of Engineering and Informatics

University of Bradford

2014

Abstract

Mohammed Reza Kianifar

Application of Multidisciplinary Design Optimisation Frameworks for Engine Mapping and Calibration

Keywords: Multidisciplinary Design Optimisation, Model Based Calibration, Design of Experiments

With ever-increasing numbers of engine actuators to calibrate within increasingly stringent emissions legislation, the engine mapping and calibration task of identifying optimal actuator settings is much more difficult. The aim of this research is to evaluate the feasibility and effectiveness of the Multidisciplinary Design Optimisation (MDO) frameworks to optimise the multi-attribute steady state engine calibration optimisation problems. Accordingly, this research is concentrated on two aspects of the steady state engine calibration optimisation: 1) development of a sequential Design of Experiment (DoE) strategy to enhance the steady state engine mapping process, and 2) application of different MDO architectures to optimally calibrate the complex engine applications. The validation of this research is based on two case studies, the mapping and calibration optimisation of a JLR AJ133 Jaguar GDI engine; and calibration optimisation of an EU6 Jaguar passenger car diesel engine. These case studies illustrated that:

- The proposed sequential DoE strategy offers a coherent framework for the engine mapping process including Screening, Model Building, and Model Validation sequences. Applying the DoE strategy for the GDI engine case study, the number of required engine test points was reduced by 30 – 50 %.

- The MDO optimisation frameworks offer an effective approach for the steady state engine calibration, delivering a considerable fuel economy benefits. For instance, the MDO/ATC calibration solution reduced the fuel consumption over NEDC drive cycle for the GDI engine case study (i.e. with single injection strategy) by 7.11%, and for the diesel engine case study by 2.5%, compared to the benchmark solutions.

Acknowledgement

I would never have been able to finish my dissertation without the support, patience and guidance of the following people; to them I owe my deepest gratitude.

First of all, I would like to thank my supervisor, Prof. Felician Campean for his excellent guidance, caring and patience. This thesis would have not been possible without his invaluable support on both academic and personal level.

I would like to thank Dave Richardson, from Jaguar Land Rover Company, for his help and technical input, not to mention his unsurpassed knowledge of internal combustion engines. Also, special thanks go to Tim Beattie from Jaguar Land Rover Company, who kindly provided the opportunity to validate the work in this thesis on a complex engine application, and Lia Beddow, CREO's project manager, for whose support and friendship I am extremely grateful.

I would like to thank Tom Patrick Dwyer, who as a colleague and a good friend was always willing to help and give his best suggestions, and also the research team at the University of Bradford's Powertrain research center, in particular to Dr. William Bradley, Prof. Kambiz Ebrahimi and Dr. Byron Mason.

I would like to acknowledge the UK Technology Strategy Board (TSB) who provided the necessary financial support for this research.

I would like to express my deepest gratitude to my parents for their unconditional love and support. They have always believed in me; and encouraged me with their best wishes.

Finally, I would like to thank my wife, for her love, support and constant patience. She has always been there to cheer me up.

Table of Contents

List of Figures

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

List of Tables

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Nomenclature

AAO	All At Once
AELH	Audze - Eglais Latin Hypercube
AFR	Air Fuel Ratio
AFS	Air Mass Flow
AIC	Akaike Information Criteria
ATC	Analytical Target Cascading
ATDC	After Top Dead Centre
BLISS	Bi-Level Integrated Synthesis
BMEP	Brake Mean Effective Pressure
BP	Boost Pressure Set Point
BSFC	Brake Specific Fuel Consumption
BTDC	Before Top Dead Centre
BWSD	Box and Wilson Steepest Descent
CO	Collaborative Optimisation
CoNcaD	Constrained Non-collapsing Design
CoV	Coefficient of Variation
CP	Columnwise-Pairwise
CREO	CO_2 Reduction Through Emissions Optimisation
CSSO	Concurrent Subspace Optimisation
DI	Double Injection
DoE	Design of Experiments
DSM	Design Structure Matrix
ECU	Electronic Control Unit
EGR	Exhaust Gas Recirculation
EMS	Engine Management Systems
EVC	Exhaust Valve Closing
FC	Fuel Consumption
FRP	Fuel Rail Pressure
GA	Genetic Algorithm
GDI	Gasoline Direct Injection
IDF	Individual Disciplinary Feasible

IQ	Post Injection Quantity
IVO	Inlet Valve Opening
KTC	Kuhn-Tucker Conditions
LH	Latin Hypercube
LM	Lagrange Multipliers
LOLA	LOcal Liner Approximations
LP	Low Pressure EGR Fraction
MBC	Model Based Calibration
MB-MV	Model Building - Model Validation
MDF	Multidisciplinary Feasible
MDO	Multidisciplinary Design Optimisation
MFB	Mass Fraction Burned
MIT	Main Injection Timing
MOO	Multi-Objective Optimisation
MSE	Mean Squared Error
NBI	Normal Boundary Intersection
NEDC	New European Drive Cycle
NLP	Non-Linear Programming
NMEP	Net Mean Effective Pressure
NN	Neural Network
NSGA-II	Elitist Non-dominated Sorting Genetic Algorithm
OLH	Optimal Latin Hypercube
PermGA	Permutation Genetic Algorithm
PEV	Predictive Error Variance
Pm	Particulate Mass
Pn	Particulate Number
PRESS	Predication Error Sum of Squares
PSO	Particle Swarm Optimisation
RBF	Radial Basis Function
RLH	Random Latin Hypercube
RMSE	Root Mean Square Error
RSM	Response Surface Modelling
SAND	Simultaneous Analysis and Design

SDNMEP	Standard Deviation of Net indicated Mean Effective Pressure
SHCB	Six-Hump-Camel-Back Function
SI	Single Injection
SLE	Successive Local Enumeration
SOI	Start of Injection
SOO	Single Objective Optimisation
SQP	Sequential Quadratic Programming
SSE	Sum of Squared Errors
ST	Spark Timing
TPLH	Translational Propagation Latin Hypercube
XDSM	Extended Design Structure Matrix

Chapter 1: Introduction

1.1 Background

Modern engines are expected to meet increasingly demanding performance criteria in response to changing legislation and consumer expectations. Advancements in engine design, such as variable valve timing, multiple injections, exhaust gas recirculation, and direct injection, aim to increase control of the combustion processes in order to meet these targets. With more engine actuators and controls, bringing additional calibration requirements for the engine electronic control unit (ECU), the engine mapping and calibration task of identifying optimal actuator settings is significantly more involved. Therefore, to address the calibration challenges, Model Based Calibration (MBC) frameworks [1]–[8] have been widely used as a way of enhancing the effectiveness of the engine calibration for both diesel and modern gasoline engines.

The MBC framework for steady state engine mapping and calibration is underpinned by the use of efficient Design of Experiments (DoE) strategies [9]–[16] to collect engine response data, typically from steady state engine tests, in order to study the effect of calibration parameters and to characterise the engine behaviour. Response surface models [17]–[19] are fitted for key engine responses (both outputs such as performance and emissions, and independent state variables such as temperatures and combustion noise), based on engine test data collected from engine dynamometer facilities. Steady state mapping aims to characterise the performance and emissions responses of the engine at specific engine speed / load operating points (often referred to as “minimap” points), typically selected to represent the engine behaviour with reference to a drive cycle (e.g. an emissions drive cycle such as NEDC [20]).

Conventional calibration optimisation based on steady state test data [21]–[24], supported by software tools such as the Model Based Calibration Toolbox™ [25], is a sequential process, iterating between “*local optimisation*”, which aims to identify optimal actuator settings at each minimap point, usually as a trade-off between conflicting objectives, and “*global optimisation*”, which aims to select a

set of local optimal solutions that meet the overall performance objectives over a target the drive cycle (e.g. minimal fuel consumption while meeting the emissions targets over the NEDC cycle) as well as calibration engineering requirements and preferences (e.g. smooth actuator maps). However, this iterative process can be a very time consuming, demanding calibration expertise in selecting a good set of global solutions, often requiring re-sampling from the local trade-off solutions set if global constraints cannot be met, and significant time and effort for the downstream calibration process.

This defines the need for better optimisation frameworks and strategies to handle the high dimensional calibration optimisation problem while addressing the complex couplings between system control variables. From an optimisation point of view the complexity of the problem is given by the large number of variables, large number of constraints, both local (at minimap level) and global (over the drive cycle), and the many objectives that have to be traded off [26].

Multidisciplinary Design Optimisation (MDO) frameworks have been introduced in literature as a more efficient approach for dealing with modern engineering systems with high and strong coupling interactions [27], which are commonplace in modern aircraft and automotive vehicles [27]–[33]. Solving such complex optimisation problems requires a methodology that can decrease the dimensionality, simplify / reduce the cost of the analysis while maintaining the consistency of the system [34]. MDO was described [35] as a methodology for the design of complex engineering systems that are governed by mutually interacting physical phenomena (subsystem or discipline) and made up of interacting subsystems or disciplines. MDO involves the development an engineering disciplinary decomposition to describe the interacting phenomena of the complex system. Several MDO approaches have been proposed to deal with practical problems of design optimisation of complex engineering systems, such as Individual and Multiple Discipline Feasible (IDF / MDF) [36], Simultaneous Analysis and Design (SAND) [37], Bi-Level Integrated Synthesis (BLISS) [38], Concurrent Subspace Optimisation (CSSO) [39], Collaborative Optimisation (CO) [32]–[35], and Analytical Target Cascading (ATC) [40]–[42].

MDO frameworks have been recently introduced for calibration optimisation problems, in particular a Collaborative Optimisation MDO framework was applied to a Diesel engine calibration optimisation [43], showing that the MDO can offer clear advantages in terms of calibration optimisation problem formulation and quality of the solutions.

1.2 Research Motivation

This research is conducted as a part of CREO project [44] (CO_2 Reduction Through Emissions Optimisation), which was one of the consortia funded by Technology Strategy Board (TSB) to run innovative projects aimed at strengthening the ultra-low carbon vehicle capability in United Kingdom. CREO project included developments in 3 technologies, with the aim of meeting legislation, customer and business requirements while minimising CO_2. These synergistic but independent technologies were:

- On-board generation and use of hydrogen, aiming to improve both combustion and after-treatment efficiency.
- Use a novel after-treatment technique, through further investigation in formulation of exhaust catalysts.
- Development and application of new optimisation tools, to improve the current optimisation methods through applying a novel optimisation technique for the engine multi-objective optimisation problem.

9 industrial and academic partners were collaborating to execute the CREO project: Land Rover, Ford Motor Company Ltd, Johnson Matthey Plc, ITM Power Ltd, Revolve Technologies Ltd, Cambustion Ltd, University of Liverpool, University of Birmingham, and University of Bradford. The role of University of Bradford in CREO project was to develop and apply a multi-attribute optimisation framework to optimise the steady state engine calibration optimisation problem. The need for a new optimisation strategy to optimise the steady state engine calibration optimisation problems has been initiated from a general observation that the steady state calibration optimisation problems are getting more complex, i.e. due to ever increasing number of calibration

parameters by development of new technologies and also the newer emission legislations which are getting harder to meet.

Quality of the steady state calibration optimisation process is directly associated with both the accuracy of approximation models and the efficiency of optimisation framework. Therefore, the work discussed in this thesis focuses on two main aspects:

- Improving the current steady state engine mapping process, aiming to improve the current design of experiment strategies in a way to deliver high fidelity response models with a fewer number of test points.
- Applying multi-attribute optimisation frameworks to solve the model based calibration optimisation problem, aiming to facilitate optimisation of highly complex steady state engine calibration problems through a better formulation of the optimisation problems.

The experimental work to validate the research was in conjunction with mapping and calibration study of AJ133 V8 naturally aspirated 5L gasoline direct injection engine (GDI), with the target of reducing CO_2 emissions while meeting forthcoming emissions legislation (EURO VI [45]) based on particulate number (Pn) emissions.

Preliminary studies have shown that Pn emissions for GDI engines is an order of magnitude higher than port fuel injection engines [46], as illustrated in Figure 1.1, which incurs a significant challenge for after-treatment. Several studies, based on single cylinder engine experiments, have investigated the engine parameters that affect Pn emissions from a GDI engine [46], [47]. The main focus of these studies has been on identifying engine control and calibration factors associated with air and fuel delivery, combustion control and engine operation, that can be used to minimise particulate formation, in a way that would reduce the need for additional after-treatment hardware (e.g. particulate filter). In this context, effective calibration optimisation plays a highly significant enabling role for the development of Pn emissions compliant engines.

Furthermore, an alternative diesel engine case study was introduced to further validate the developed multi-attribute optimisation strategies to optimise the steady state calibration problems.

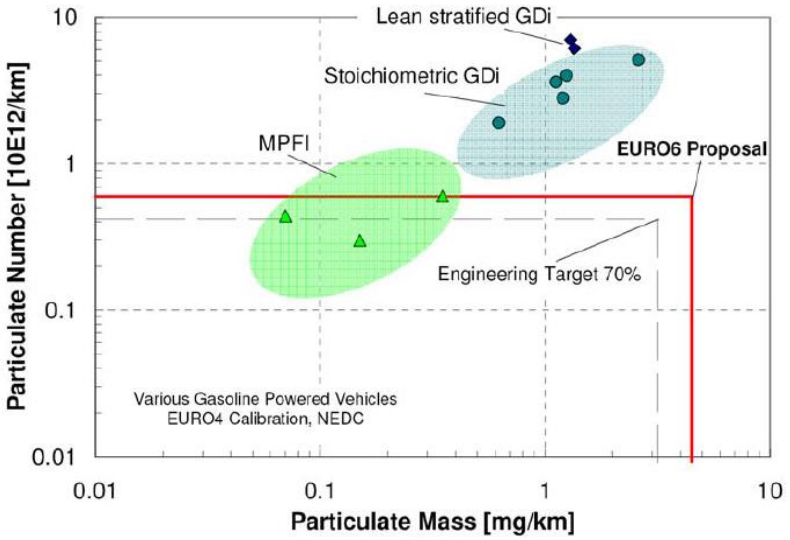


Figure 1.1: Vehicle emission proposal for EURO VI [46]

1.3 Research Objectives

The following objectives were fulfilled to accomplish the research targets:

- To develop an efficient design of experiment strategy, in order to minimise the number of required test points to achieve a response model of target accuracy without demanding prior knowledge about the system behaviour or model type.
- To demonstrate the application of developed design of experiment strategy for the steady state engine mapping problems.
- To develop different MDO optimisation frameworks to formulate the steady state engine calibration optimisation problems, aiming to improve the problem formulation of the highly complex steady state engine calibration optimisation problems using a logical problem decomposition strategy.
- To demonstrate the application of developed MDO frameworks for the steady state engine calibration optimisation problems.

1.4 Thesis Organisation

The thesis commences by analysing steady state engine model based calibration process in Chapter 2, including a broad discussion of different design of experiment strategies and different statistical methods of response surface modelling, followed by reviewing of currently used experimental design and modelling techniques for steady state engine mapping process.

Chapter 3 reviews the existing optimisation techniques in three main categories: ‘Single Objective Optimisation algorithms’, ‘Multi-Objective Optimisation algorithms’, and ‘Multidisciplinary Optimisation algorithms’, with a focus on the latter category. In this section, the state of art of each optimisation category is presented, and their performances is analysed in details. This chapter also covers a sound review of the optimisation techniques have been applied for the steady state engine calibration problems.

Chapter 4 presents the research methodology planned to satisfy the research objectives. This chapter also explains the engine case studies, data acquisition system, and the commercial software packages used to conduct the research.

Chapter 5 presents some original work by development of a sequential design of experiment strategy, in order to improve the testing efficiency through reducing the number of test points required to achieve an accurate response surface model. The proposed strategy is principled on extension of Optimal Latin Hypercube designs, which is a space-filling design, with the aim of distributing the points evenly within entire design space without a particular assumption about the model type or the number of required sample points. The developed sequential DoE strategy is then applied for a GDI engine case study, to fit the required approximation models and validate the behaviour of engine responses.

Chapter 6 describes application of different multidisciplinary optimisation frameworks to formulate the steady state engine calibration problems, and studies the effectiveness of these optimisation strategies for the GDI engine case study. For further validation, the developed optimisation frameworks are

also applied for an alternative diesel engine case study, provided by the sponsoring company.

Finally, Chapter 7 summarises the discussions, and discusses the final conclusions, research original contributions, and recommendations for future work.

An outlook of how to read this thesis is given in Figure 1.2. Chapter 1 discusses the research motivation, aims and objectives. Then, revision of current literature regarding steady state engine mapping, including design of experiment and response surface modelling techniques, is presented in chapter 2. This is followed by a sound review of available optimisation techniques presented in chapter 3. Chapter 4 explains the engine case studies used in this research and the way research is conducted throughout the thesis. Chapter 5 is intended to explain development of an original sequential design of experiment strategy, which is implemented for the GDI engine case study to deliver high fidelity response models. Chapter 6 discusses implementation of several multidisciplinary optimisation frameworks to the GDI engine case study, and then further validation of the methods on a diesel engine application. Finally, the research findings are discussed in Chapter 7, followed by suggesting further research opportunities in this area.

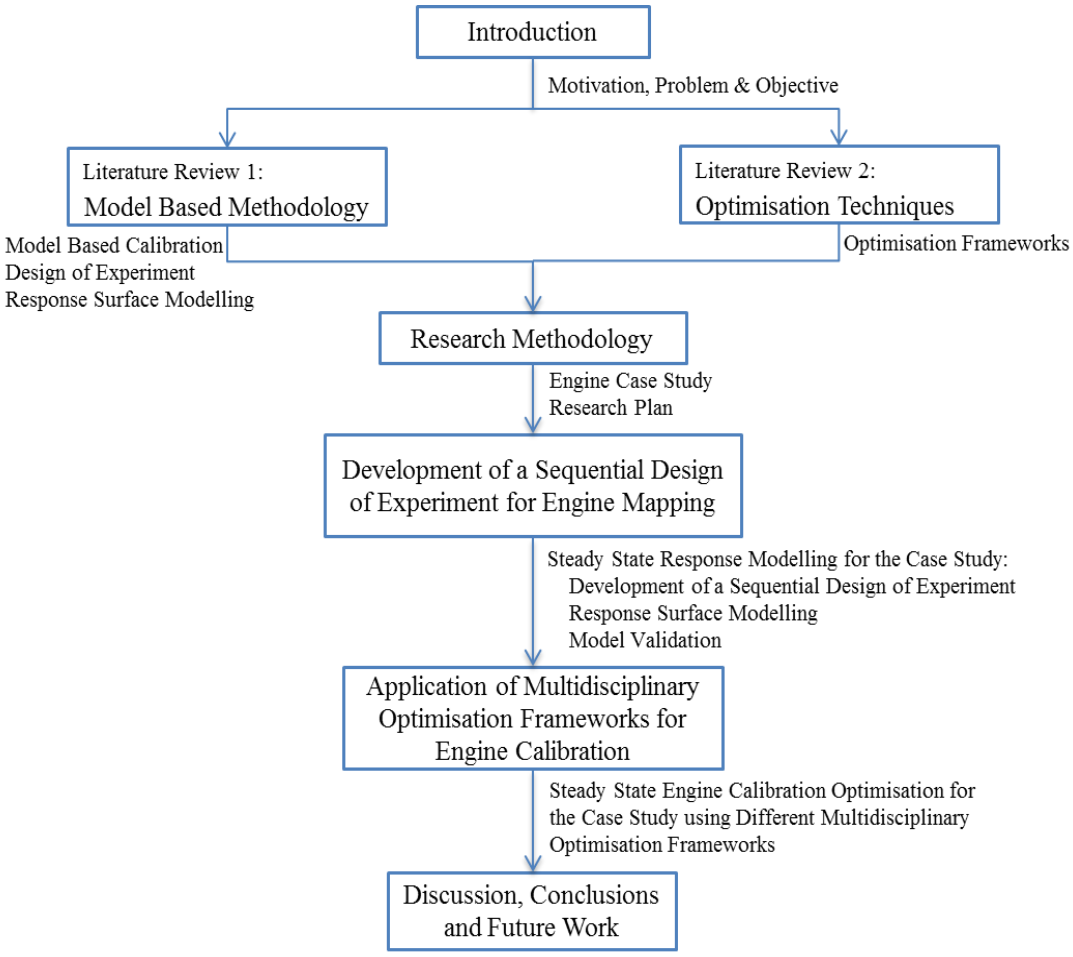


Figure 1.2: Flowchart for reading this thesis

Chapter 2: Review of Engine Mapping Methodology

2.1 Introduction

In this chapter a broad review of literature is presented on existing design of experiment strategies and statistical modelling techniques, with a focus on their application for steady state engine model based calibration. This survey is presented in four sections:

- Section 2.2 presents Model Based Calibration (MBC) process.
- Section 2.3 covers an overview of existing DoE strategies, including both standard single-level and sequential DoE strategies.
- Section 2.4 analyses the most commonly used modelling methods in engine mapping, including both parametric and non-parametric modelling techniques.
- Section 2.5 reviews application of discussed DoE methods and modelling techniques in engine model based calibration area.

2.2 Overview of steady state Model Based Calibration (MBC)

2.2.1 Electronic Control Unit (ECU)

In an internal combustion engine (ICE) all the control parameters and sensors are embedded in the engine management systems (EMS). The Electronic Control Unit (ECU) is a component of the engine management system which is designed to control the state of the powertrain through controlling all the engine sensors and actuators [48]. The ECU uses sensors and processing algorithms to detect the engine operating conditions. Subsequently, the ECU uses this information to send appropriate signals into the actuators to set the engine at the desired state. These signals are provided by the look up tables stored in the ECU as the basis of control strategy. Look up tables are two dimensional arrays containing information about engine desire state and actuation parameters [49].

As an example of how the ECU works, assume a situation where a driver decides to accelerate by pressing the accelerator pedal [49]. To fulfil the driver's

acceleration demand, accelerator-pedal sensor sends information to the ECU to define the pedal position. Accordingly, the ECU sends a signal to set the throttle valve (i.e. based on the ECU table) in order to obtain the correct cylinder air charge corresponding to the pedal position.

ECU tables are generated through calibrating the prediction models developed during the engine mapping process [48]. The Engine mapping process involves fitting statistical models to a set of engine data, which is collected with the engine data acquisition system, in order to model the true behaviour of engine in relation with the independent input parameters [50], [51]. In the following section, steps of engine mapping process are explained in detail.

2.2.2 Engine Model Based Calibration Process

The development of engine technologies to improve the performance, fuel economy and drivability while meeting increasingly stringent emissions legislation has resulted in an increased complexity of powertrain calibration with significant time and cost implications. One of the main challenges in the calibration process is to find the best combination of engine parameters to compromise between the conflicting engine calibration requirements, such as emissions versus fuel consumption.

The traditional calibration strategies are focused on optimising the engine parameters one-at-a-time on the test bed [52]. The one-at-a-time strategy can be an effective approach under certain conditions, such as the times that parameters are independent, the number of runs is limited and the aim is to achieve improvements, and when the testing experimental error is relatively insignificant compared to the parameter effects [53]. However, this strategy not only ignores the interaction between the parameters, but also is very time consuming and inefficient for complex engine calibration problems (i.e. with many parameters to optimise). Therefore, to overcome this calibration challenge with satisfactory expenditure of costs and time, strategies such as model based calibration (MBC) have been introduced [50], [51]. A Model based calibration strategy empowers calibration engineers to optimise all the engine control parameters simultaneously.

Steady state model based calibration is based on statistical modelling of the engine responses of interest using the test data collected at fixed speed/load operating points. The engine response models are required to be of sufficient fidelity to approximate the engine behaviour with an acceptable accuracy. A steady state calibration is derived through mathematical optimisation of the engine response models [50], [51]. The standard steady state model based calibration (MBC) process is shown in Figure 2.1.

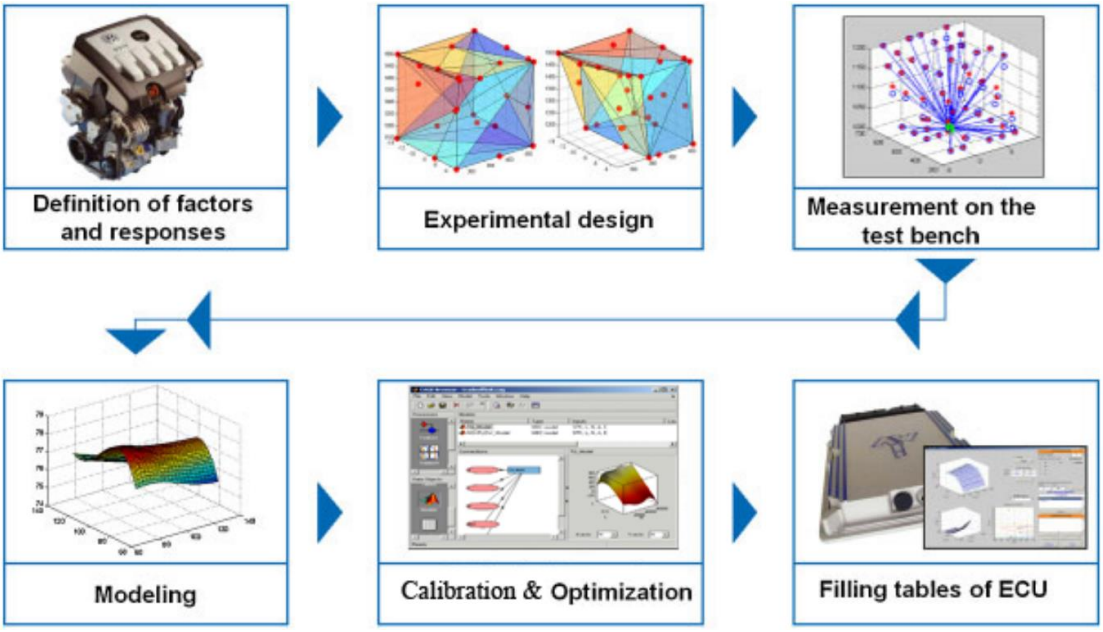


Figure 2.1: Steps of model based calibration [49]

The standard workflow of steady state model based calibration process (also known as Z-process [7]) is as following [54]:

1) **Definition of factors and responses:** The first stage of steady state testing is to determine the independent controllable engine parameters and engine responses of interest at a number of speed / load operating points (also called minimap points). Therefore, the base calibration will rely on a number of local models at different specific minimap points [55].

   The number and location of minimap operating points within the speed / load envelope are selected by calibration engineers, to be representative of a whole emission drive cycle. The purpose of emission legislations is to regulate exhaust emission limits for a typical driving profile. As an example, New European Drive Cycle (NEDC) driving profile consists of

four repeats of low speed urban driving cycle and a highway driving cycle, within 1186 seconds of testing duration [20], as illustrated in Figure 2.2.

2) **Experimental design:** Design of Experiment (DoE) methods have been widely applied to enhance the effectiveness of the steady state testing. In simple terms the main purpose of DoE methods is to collect the maximum possible information with the least measurement effort [51].

3) **Measurement on the test bench:** Steady state data is usually collected at the designed DoE test points (at each minimap point) in a dynamometer cell equipped with data acquisition system.

4) **Modelling:** The test bench measurements are used to fit a type of statistical model which can predict the engine response behaviour accurately over the entire range of actuators [56].

5) **Calibration and optimisation:** The developed prediction models can be used during the calibration process to find the optimum operational settings for the actuators by implementing an optimisation technique.

6) **Filling the ECU tables:** The optimum settings obtained from the calibration process are inserted to the ECU tables over the entire range of engine speed / load points.

These six stages are performed in succession and can be directed offline or online, regarding the test bench automation [57], [8].

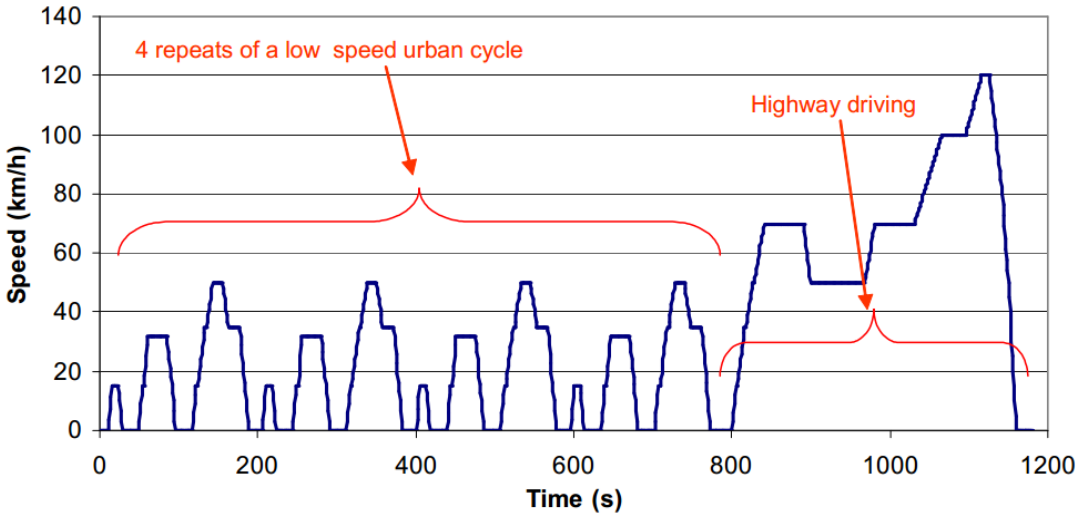


Figure 2.2: An example of NEDC drive cycle [20]

Steady state model based calibration has been employed for many engine applications, for both gasoline and diesel engines, e.g. optimisation of catalyst [58], engine map optimisation [1], calibration of camshaft timing for a direct injection engine [22]. However, considering the fact that steady state calibration process in gasoline engine requires data collection at more than 30 engine speed / load conditions, more than the diesel engine, engine testing is still expensive [59]. To overcome this problem, engine global models have been introduced [60], [61]. The main idea in global modelling approach is to have one global model for engine behaviour over the entire speed / load range instead of having several engine local models (for each speed / load setting) [62]. In this model, speed and load are also considered as variables in the design of experiment (DOE) step.

Considering that changing engine speed / load operating conditions have strong effects on engine combustion, global models typically experience an unacceptable loss in model accuracy, which might result in inaccurate engine models for optimisation purposes [63]. Therefore, both *global surrogate* models and *local surrogate* models are employed for specific applications. In general, local surrogate models are employed to guide the optimisation algorithm towards the global optimum, while, the global surrogate models are used an alternative to the simulator, to mimic the whole simulator/ or system behaviour within an acceptable accuracy over the entire design domain [64].

Figure 2.3 illustrates an example of fitting a response surface model to a set of DoE test points. In this figure, the designed experimental test points are collected by running the physical simulator, and then the simulator outputs (response values) are used to fit a proper response surface model. It can be argued that both choice of data points (i.e. location of the test points within the design space) and type of statistical models have a viable effect on the accuracy of a surrogate model [65]. Therefore, a comprehensive review of existing DoE strategies will be presented in the next section, followed by analysis of some of the frequently used statistical models in engine model based calibration area.

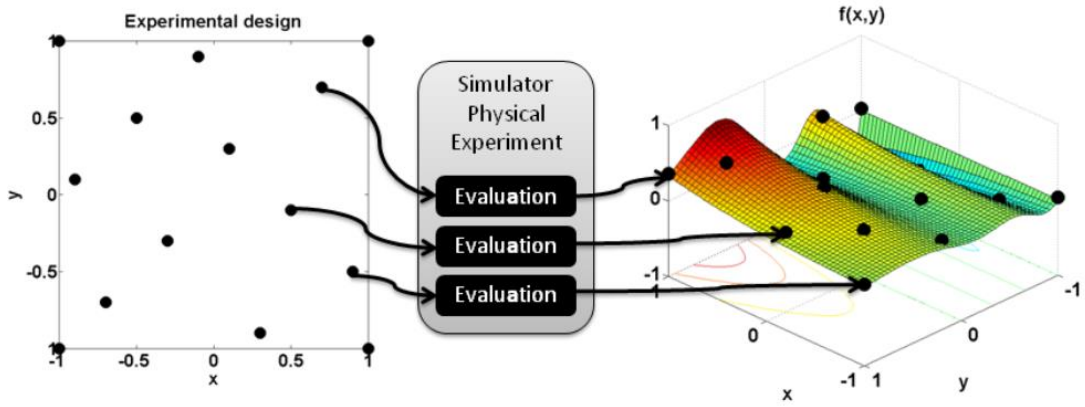


Figure 2.3: Fitting a surrogate model to the evaluated data points in the simulator [65]

2.3 Design of Experiment (DoE) Methods

The Design of Experiment (DoE) method is used to determine a set of test points in order to derive enough information to describe behaviour of a response over whole range of input variables from the minimum number of test points [66]. DoE was originally introduced in agricultural experiments by Sir Ronald A. Fischer [67] in early 1920's. His pioneering work in introducing statistical principles (i.e. randomisation, replication, and blocking) into designing an experiment has led into further research in DoE area, and subsequent DoE developments.

Although DoE is not a new concept, application of DoE techniques in the automotive industry, i.e. particularly in engine development, has become common practice for a little more than a decade [49], e.g. optimisation of catalyst system [61], calibration of camshaft timings to optimise engine performance [4], [22], and many other applications. The main reasons for popularity of DoE techniques in automotive industry are:

- Development of response surface methodology by Box and Wilson [68] that sparked a new era in statistical design.
- Further research in developing more advanced statistical DoE techniques such as the work of Taguchi [69] in late 1970s (e.g. fractioned factorial designs).

In this section, existing DoE approaches in literature are presented in two main categories:

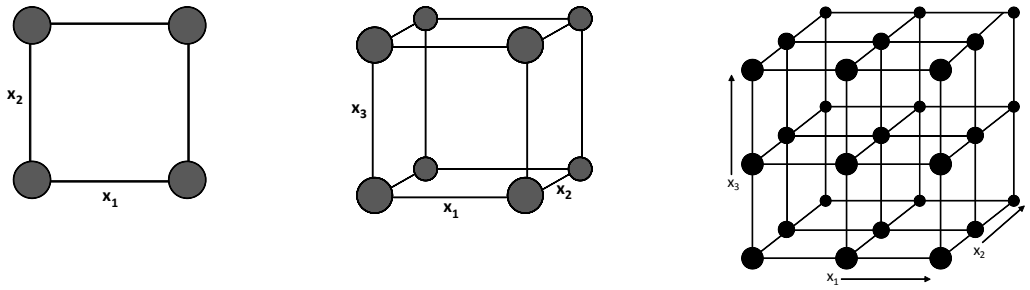
- **Single level DoE strategies** (or one-shot DoEs): in this category, all the DoE test points are collected in a single attempt. Single level DoE strategies can be further reviewed in three principle categories:
    - Classical designs
    - Optimal designs
    - Space-filling designs
- **Sequential DoE strategies** (or adaptive DoEs): in this category, the DoE test points are collected through a number of iterations. Sequential DoE strategies can be divided into two main categories:
    - Optimal sequential designs
    - Evolutionary sequential designs

2.3.1 Classical Design of Experiments

Classical designs focus on defining a set of test points to minimise the effect of random errors in physical experiments on acceptance of a model hypothesis [49]. These designs are well investigated for simple regions (e.g. hypercube) and are typically employed for simple responses that can be explained by low order polynomials. Classical experimental designs can be presented within two categories based on their structure:

- **Full factorial design:** In this design, all level combinations of the variables are equally important [70]. To plan a full factorial design for a problem with (n) variables and (k) levels, the number of required test points is (k^n) [71]. Some examples of full factorial DoE are shown in Figure 2.4. Full factorial designs are one of the primary DoE methods which were used frequently for engine testing, mainly based on polynomial models [14].
- **Fractional factorial design:** This design is a subset of full factorial design, or in other words is a subset of all level combinations of the variables. Accordingly, fractional factorial designs can be used as an alternative for full factorial designs were all the combinations of the variables are not needed, or higher order of interactions are negligible [14], [60]. Two of the broadly used fractional factorials in automotive industry [15], [61] are:

- Central composite designs: this fractional design is usually used for the problems that additional star and centre test points can augment the base design [14], as shown in Figure 2.5 (a) for an example with 3 variables / 3 levels.
- Box-Behnken designs: this design is usually implemented for the problems where the smallest number of factor levels is required [49], as illustrated in Figure 2.5 (b) for an example with 3 variables / 3 levels.

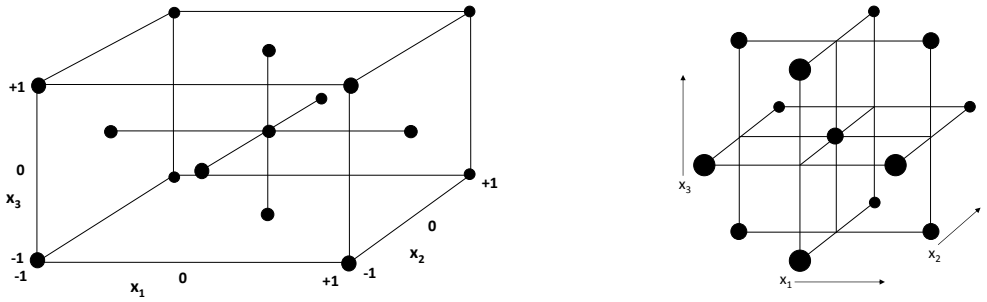


(a) 2 levels / 2 Variables (2^2 = 4 tests, linear model)

(b) 2 levels / 3 Variables (2^3 = 8 tests, linear model)

(c) 3 levels / 3 Variables (3^3 = 27 tests, quadratic model)

Figure 2.4: Three examples for full factorial designs



(a) Central Composite

(b) Box-Behnken

Figure 2.5: An example of fractional designs for 3 levels / 3 Variables

Although classical designs perform very well for simple problems (i.e. small number of variables and levels), these designs have some weaknesses such as following:

- For a larger number of variables and variable levels, the number of required sample points increases exponentially [17]. For instance, for a linear model (2 levels) with five variables, 2^5 (or 32) test points are

required to plan a full factorial DoE. However, this number increases to 64 (2^6) by adding another variable.

- Classical designs are not flexible, in a sense that all the designed test points must be feasible otherwise the final design quality might be compromised. Therefore, to implement a classical design, all design boundaries must be identified in advance to ensure all test points are valid. It is noteworthy that defining the boundary limits might require a large amount of pretesting (i.e. called screening) [49].
- Classical designs cannot be adopted for design variables with asymmetric boundary limits [71].
- For implementing a fractional factorial design instead of a full factorial design, prior knowledge about the response behaviour is needed [14].

Given these drawbacks, classical designs might not provide an efficient DoE technique for complex multi-dimensional nonlinear systems, such as engine applications [15].

2.3.2 Optimal Design of Experiments

Research into the optimality of DoEs is going back to 1960 [16]. In optimal experimental designs, the optimality of a design is associated with the mathematical properties of the pre-specified model type, and is assessed with respect to a statistical criterion [16], [19].

By assuming that the vector of observations *Y* (i.e. test outputs) can be explained by [19]:

$$Y = X * B + e \qquad \text{Equation 2.1}$$

where *X* is the matrix of design variables at DoE test points, *B* indicates the vector of model tuning parameters, and *e* defines the error vector. *B* can be estimated by using least-squares method as illustrated in the following equation [19]:

$$B = (X^T * X)^{-1} X^T Y \qquad \text{Equation 2.2}$$

In this equation, $(X^T * X)^{-1}$ is known as the variance matrix, which is defined by the inverse of the *information matrix* (IM). Different optimal designs can be obtained by optimising some functional on Equation 2.2, which are principally invariants of information matrix. Table 2.1 summarises the optimality criteria of some of the commonly used optimal designs [60], [72].

Optimal Design	Optimisation Criterion	Statistical Meaning
A-Optimal	Maximising trace of the IM	Minimising the average variance of the estimates of the regression coefficients
D-Optimal	Maximising determinant of the IM	Minimising the covariance of the parameter estimates
G-Optimal	Minimising the maximum entry in the diagonal of hat matrix or $X\ (X^T * X)^{-1}\ X^T$	Minimising the maximum variance of the predicted values
V-Optimal	Minimising the average of diagonal entries of hat matrix	Minimising the average prediction variance over the design points

Table 2.1: Optimality criterion of optimal designs

Figure 2.6 shows an example of optimal designs for a 3 variables / 3 levels problem, using a D-optimal experimental plan.

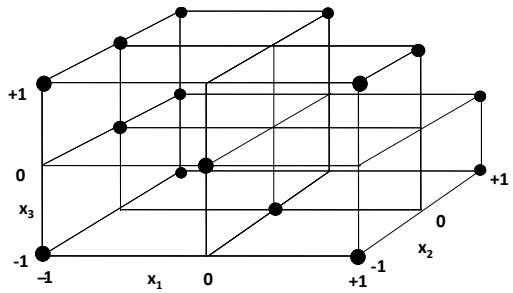


Figure 2.6: D-Optimal design for 3 levels / 3 Variables

Optimal designs have been widely implemented for steady state engine mapping applications, such as calibration of a gasoline engine camshaft timing [2], calibration of a diesel engine injectors to minimise emissions [5], and calibration of a gas exchange model [3]. Compared to the classical designs, optimal designs offer several advantages such as [60]:

- Optimal designs require less experimental runs.

- Optimal designs can accommodate various parameter types (e.g. discrete, process and mixture parameters).
- Optimal designs can be optimised for a constrained design space.

However, efficiency of optimal designs can be compromised for complex problems since [73], [74]:

- Optimal designs require prior knowledge about the model type.
- Optimal designs require prior knowledge regarding the number of test points.
- Optimal designs are not flexible and dropping infeasible test points degrades the design quality.

Accordingly, optimal designs might require a large amount of pretesting for unknown engineering problems, to define the boundary limits and estimate the response behaviour in advance to the final DoE planning [71].

2.3.3 Space-Filling Design of Experiments

Space-filling designs aim to distribute the experimental test points uniformly through the design space to collect information regarding the system behaviour from all over the design space [9]–[13]. Space-filling sampling strategy has its origins in the computer science field and firstly used for planning computer experiments [75]. Given that space-filing designs plan the experiment regardless of model type or system behaviour, these designs have been frequently used as an attractive experimental design option for steady state engine mapping problems, i.e. particularly when system behaviour is unknown [9], [11], [12]. Using a space-filling design has some advantages over the other DoE techniques, such as:

- Space-filling designs provide the opportunity to explore all parameters domain.
- Space-filling designs empower the user to employ more advanced statistical modelling techniques.

- Space-filling designs are more flexible than both classical and optimal designs. Thus, in the cases that not all the design points are fulfilling the constraint requirements, removing the infeasible points does not degrade the experimental design integrity [76]. Accordingly, to implement a space-filling design, no extensive pretesting stage is demanded.

2.3.3.1 Space-Filling Design Properties

Efficiency of a space-filling DoE technique is directly dependent on ability of the DoE technique to fulfil several statistical requirements. The main statistical requirements to have a good space-filling experimental design are discussed below:

a) Space-fillingness:

It is a crucial principle for a space-filling DoE technique to distribute the sample points evenly within the design space, regardless of the problem dimension and sample size [77]–[81]. A method to investigate space-fillingness of an experimental design is to examine the *Euclidian distance* (also called intersite distance [65]) for each sample point (x_i), which practically determines the distance between the test point (x_i) to the closest test point within the design space [72].

Equation 2.3 calculates the Euclidian distance for each sample point (D_i).

$$D_i = minimum_{\mathbf{x_i}, \mathbf{x_j} \in N} \sqrt{\sum_{k=1}^{K} \left| x_i^k - x_j^k \right|^2} \quad \text{where } i \neq j \qquad \text{Equation 2.3}$$

Where N is the sample size and K denotes the number of design parameters (or problem dimension). Therefore, a larger average of Euclidian distances for a design indicates a better distribution of test points within the design space.

b) Projective properties:

Another important requirement for a good space-filling DoE strategy is to guarantee a good projective property, which is also referred to as non-collapsingness [77]. A non-collapsing design guarantees that no two sample points project onto each other along any of the axes when the K-dimensional sample points are projected into the (K-1)-dimensional space. In other words, in

a non-collapsing design each sample point has a unique value along any of the axes [77]. Accordingly, projection criterion ensures that every parameter is represented over its domain, even if the response is only dominated by a few of the parameters [19]. Projection distance (*P*) for each sample point (x_i) along a dimension (*k*) can be calculated by the following equation [80]:

$$P_i = minimum_{\mathbf{x_i},\mathbf{x_j}\in N} \sqrt{\left|x_i^k - x_j^k\right|^2} \qquad \text{where } i \neq j \qquad \text{Equation 2.4}$$

So, a larger projection distance along an axis indicates a better distribution of test points within the axis domain.

It is argued by Stinstra *et al*. [76] that a good space-filling design does not necessarily ensure a good projection design. They claimed that when a DoE strategy is fully focused on maintaining the space-filling criterion, it often results in a collapsing design.

Using a non-collapsing design is more essential for pretesting stage (i.e. screening stage) since the main objective of screening stage is to explore the response behaviour over all parameters domain. Consequently, influential parameters on the response can be determined in screening stage by evaluating the response values of projected points on any of the axes. If a parameter is not effective on the response, the projected points along the corresponding axis will have the same response value [65].

c) Orthogonality:

The next principle to have an efficient space-filling DoE strategy is to maintain the experimental design orthogonality. This criterion assures that there is no correlation between each combination of input parameters [82], [83], thus ensuring that the experimental design is a good representative of the real variability [49]. Taking into the consideration that only a few existing experimental designs are orthogonal (e.g. factorial designs) most of the existing space-filling strategies try to reasonably satisfy the orthogonality criterion. Correlation (*r*) between each two vectors of parameters can be calculated by [79]:

$$r = \frac{\sum_i^N (x_i^k - \overline{x^k})(x_i^j - \overline{x^j})}{\sqrt{\sum_i^N (x_i^k - \overline{x^k})^2}\sqrt{\sum_i^N (x_i^j - \overline{x^j})^2}} \quad \text{where } k \neq j \qquad \text{Equation 2.5}$$

Where *N* defines the sample size, *k* and *j* indicate the dimension (or parameter), and *i* is the sample number. So, x_i^k represent the sample *i* along dimension *k* and $\overline{x^k}$ gives the average of all *N* sample points along dimension *k*. The output of Equation 2.5 is used to measure the strength of correlation between any two vectors of parameters, where $r = 1$ means that there is a perfect positive correlation between two vectors and $r = -1$ means that there is a perfect negative correlation between two vectors. Therefore, for a perfect orthogonal design (e.g. factorial designs), the correlation between any two vectors of parameters is zero.

It is noteworthy that satisfying the orthogonality criterion is not always possible. For instance, for a design with a large number of sample points (*N*) and a relatively small number of parameters (*K*), it is more likely to have a higher correlation between the vectors of variables [84].

In general, the importance of satisfying each of the discussed requirements to have an efficient space-filling DoE strategy (i.e. space-fillingness, non-collapsingness, and orthogonality) can be variant based on the engineering problem requirements. Thus, an efficient space-filling DoE strategy can be selected by considering the problem requirements.

Therefore, in the following section some of the well-known space-filling DoE techniques are reviewed within two main categories:

I. Space-filling DoEs for symmetric design spaces: these DoE strategies are principally proposed for problems with symmetric design spaces.

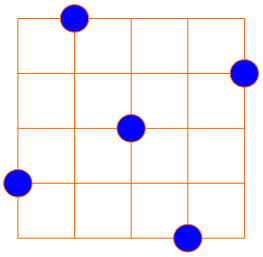
II. Space-filling DoEs for asymmetric design spaces: these DoE strategies are introduced for problems with constrained design spaces.

2.3.3.2 Space-Filling DoEs for Symmetric Design Spaces

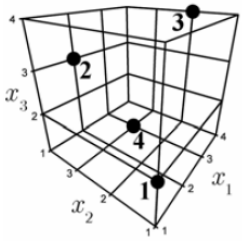
Many methods of space-filling designs have been introduced in literature for symmetric design spaces, such as Sphere Packing, Uniform design, Minimum

Potential and Latin Hypercube [85]. However, Latin Hypercube (LH) design is the most commonly used space-filling strategy in steady state engine mapping area [1], [22], [58]. The main reason for popularity of LH DoE technique is the unique property of the design, which is a compromise between spread of points (i.e. space-fillingness) and uniform spacing (i.e. non-collapsingness).

Latin Hypercube sampling strategy was originally proposed by McKay *et al*. [86] to be applied for computer experiments [87]. A LH design is generated by gridding the design space of each parameter into *N* (i.e. sample size) equidistant levels, and selecting only one test point on each level. Therefore, a LH design ensures that all levels of each parameter are represented over its range by maintaining non-collapsingness [75]. Figure 2.7 illustrates an example of LH designs for two simple cases: (a) for a 2-dimensional problem with 5 sample points, and (b) for a 3-dimensional problem with 4 sample points.



Point	x_1	x_2	x_3
1	1	1	2
2	2	4	3
3	4	2	4
4	3	3	1



(a) for 5 sample points/2 Variables (b) 4 sample points/3 Variables [10]

Figure 2.7: Latin Hypercube (LH) designs

LH sampling strategy has many advantages over other space-filling designs [75], [88], such as:

- Maintaining a good projection property.
- Fast computation.
- Easy implementation.
- Dealing with complex engineering problems (i.e. with a large sample sizes and/or a large number of parameters).

However, LH design does not always result in an acceptable experimental plan since this design does not necessarily reflect space-fillingness and orthogonality requirements. For instance, Figure 2.8 illustrates an example of LH design with

both good and poor design properties. In this figure, design (a) shows a good LH design in which test points are distributed evenly within design space without a certain correlation between the parameters, however; design (b) satisfies neither of space-fillingness nor orthogonality criteria.

Therefore, many approaches have been proposed in literature to improve the LH experimental strategy. These approaches can be divided into two main categories: ‘non-optimisation based’ approaches, and ‘optimisation based’ approaches.

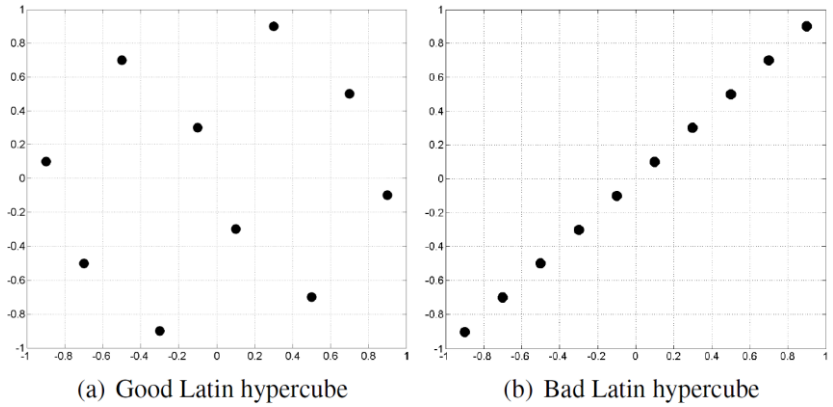


Figure 2.8: An example of LH designs with different space-filling properties [89]

1. **Non-optimisation based approaches**

Several non-optimisation based methods have been proposed to enhance the LH design properties (i.e. space-fillingness and orthogonality) by modifying the sampling technique. Some of these methods are reviewed as below:

Tang [82] proposed an approach called orthogonal array-based LH, to reduce variance of the sample points mean by evenly sampling of whole design space. However, this approach is computationally expensive, since to produce an orthogonal design all the test points must be generated simultaneously.

Steiberg and Dennis [90] proposed an algorithm established on the idea of generating a LH design with good orthogonal properties as rotations of factorial designs. This design showed good statistical properties for high dimensional problems (more than 10 variables). However, this strategy is not flexible regarding the model type. Also, considering that the final design in this strategy

is derived from a full-factorial design, the number of total sample size is not flexible.

Vianal *et al*. [91] proposed a pattern based algorithm, called Translational Propagation algorithm (TPLH) [91], [92], to generate a LH design with good intersite distance properties (i.e. intersite distance indicates the design space-fillingness). This strategy is based on propagation of small building blocks, called seeds, in hyperspace. These seeds consist of one or more points each, and are used to recreate the same patterns throughout the design space. TPLH is a very fast approach, however; the main shortcomings of TPLH algorithm are:

1) High correlation between pairs of variables for high dimensional problems. Poor orthogonality property results in complication of data analysis process, and difficulty in defining the variables with higher effects on the outputs [90].
2) Poor space-filling property for high dimensional problems (more than 6 variables).

Successive Local Enumeration (SLE) algorithm is also a non-optimisation based approach proposed for generating one-shot LH DoEs [93]. This algorithm is established on the idea of selecting a random initial sample point, and proceeding the sampling process iteratively by maximising the minimum of all the distances between the additional sample point and the existing sample points. In this strategy, the total number of sample points is required to grid the design space, and allocate the points to each subsection one-by-one. The final DoE generated by this strategy can guarantee good intersite and projected distances, however; the quality of final design can be affected by the location of initial sample point. Also, this design is not an efficient approach for problems with more than 5 variables due to computation expenses and poor space-fillingness.

Cioppa and Lucas [84] developed an algorithm to generate LH designs for high dimensional problems, in a way to satisfy both space-fillingness and orthogonality demands. This algorithm is based on generating several LH designs and calculating the orthogonality criteria for each of them. Then,

designs that pass the requirement for minimum allowable correlation among the columns of variables will be ranked based on their intersite distance. The best design will be selected as the final design. Given that generating several LH designs, and calculating the corresponding intersite distance and correlation is time consuming, this process is computationally inefficient for high dimensional problems.

2. Optimisation based approaches

An alternative idea to improve LH design is to generate Optimal Latin Hypercube (OLH) designs by adopting some optimality criterion for LH design [49]. An OLH design can be generated by either enumeration of possible combinations of parameters or an optimisation approach. Enumeration is not practical for complex problems since considering all possible combinations of variables is expensive and time consuming. For example, for a simple problem of 10 sample points and 5 variables, there are 6×10^{32} possible combinations. If each solution takes one nanosecond (1×10^{-9} s) to evaluate, the whole evaluation process would take approximately 2×10^{16} years [94], which is clearly infeasible. Therefore, distribution of the sample points for a complex OLH design can be conducted as a discrete optimisation problem [10]. The main challenges for implementing a discrete optimisation problem are:

a) Select an optimality criterion: to define the objective function.

b) Implement an efficient optimisation algorithm: to generate the most representative experimental design.

a) Select an optimality criterion:

Table 2.2 gives an overview of the frequently employed optimality criteria to generate an OLH design with a good space-filling property. Of the reviewed optimality criteria in Table 2.2, both Maximin [77]–[81] (i.e. maximising the minimum distance between every two samples) and Audze Eglais [95] (i.e. minimizing the system potential energy through maximising the distance between every two points) functions has proven to maintain a good intersite distance. It is discussed by Draguljić [96] that Audze Eglais performs better for high dimensional problems (K). Maximin criteria tends to generate more sample points around the corners, especially for high dimensional problems [96],

consequently this strategy might not preserve good space-filling properties at the centre of design space, particularly for a small sample size (*N*).

The $\emptyset_p$ [80] criterion is an extended version of Maximin equation. In this formulation, given in Table 2.2, if *p* is a large number, this criterion behaves comparable to the Maximin equation, while for a small *p* value, $\emptyset_p$ performs similar to the Audze Eglais criterion. However, implementation of $\emptyset_p$ criterion is not always straight forward since this equation has shown to be numerically unstable in certain situations [65]. For instance, for a large *p* value, it might be the case that $\emptyset_p$ equation returns an infinity value when two points are too close to each other. In such cases, even one intersite distance with zero value can result in an infinity $\emptyset_p$ value. Furthermore, the $\emptyset_p$ value doesn't have a geometrical meaning (i.e. it is a comparative ranking between different designs), therefore it is difficult to combine $\emptyset_p$ optimality criterion with other design requirements (such as orthogonality). Additionally, as a consequence of the extreme $\emptyset_p$ values, it is a challenging task to visualise the optimisation surface [91].

Optimality Criterion	**Formula**
Manhattan (l_1 norm) [77], [78]	$min_{\mathbf{x_i},\mathbf{x_j}\in N} \sum_{k=1}^{K}\left\|x_i^k - x_j^k\right\|$
Maximin (l_2 norm) [77]–[81]	$min_{\mathbf{x_i},\mathbf{x_j}\in N} \sqrt{\sum_{k=1}^{K}\left\|x_i^k - x_j^k\right\|^2}$
Audze Eglais [95]	$\sum_{\mathbf{x_i},\mathbf{x_j}\in N} \sqrt{\sum_{k=1}^{K}\left\|x_i^k - x_j^k\right\|^2}$
$\emptyset_p$ [80]	$\left(\sum_{\mathbf{x_i},\mathbf{x_j}\in N} \sqrt{\sum_{k=1}^{K}\left\|x_i^k - x_j^k\right\|^2}^{p}\right)^{1/p}$

Table 2.2: Overview of different space-filling criteria

b) Implement an efficient optimisation algorithm:

Different global optimisation algorithms have been proposed in literature for generating OLH designs, such as columnwise-pairwise, simulated annealing, and Permutation Genetic Algorithm (PermGA) [9]–[11], [97]. It has been argued in literature that PermGA perform more efficiently for higher-dimensional OLH DoE problems (more than 5 variables) due to a convergence rate corresponding

to the varying number of actuators [10], [11]. In other words, PermGA can generate better distributed points for high dimensional DoEs while reducing the computational costs. Liefvendahl [11] also discussed that Columnwise-pairwise (CP) algorithm is performing more efficiently for small and medium LH designs.

Table 2.3 summarises some examples of employed optimality criteria and optimisation algorithms to develop an OLH design.

Author	**Optimisation Algorithm**	**Objective Function**
Audze and Eglais [97]	Coordinates exchange algorithm	Audze Eglais
Morris and Mitchel [80]	Simulated annealing	$\emptyset_p$
Ye *et al.* [78]	Columnwise-Pairwise	$\emptyset_p$
Bates *et al.* [9]	PermGA	Audze Eglais
Bates *et al.* [10]	PermGA / Simulated annealing	Audze Eglais
Jin *et al.* [98]	Enhanced stochastic evolutionary algorithm	$\emptyset_p$
Liefvendahl *et al.* [11]	PermGA / Columnwise-Pairwise	Audze Eglais / Maximin
Van Dam *et al.* [77]	Branch-and-bound algorithm	Manhattan (l_1 norm)

Table 2.3: Examples of the introduced techniques in literature to construct an OLH design

The examples presented in Table 2.3 are principally focused on delivering an OLH design with good space-fillingness property. However, Joseph and Hung [79] argued that although it is expected to have a good space-filling LH design when the correlation between parameters is minimized, due to the intuitive relation between minimizing orthogonality and maximizing the intersite distance [82], [83], there is no direct, one-to-one, relationship between these two criteria [79]. So, to deliver a good OLH design, Joseph and Hung [79] suggested to combine both space-fillingness and orthogonality criteria using a multi-objective optimisation function. They used simulated annealing optimisation algorithm as the solver of optimisation problem. However, the main issue of the developed algorithm is the computation efficiency, especially for high dimensional problems.

In general, the biggest drawback of LH designs compared to classical and optimal DoE strategies, for both 'optimisation based' and 'non-optimisation based', is that LH design is not fully capable of predicting the response values at the points outside the area where the data is collected. Accordingly, for LH designs the prediction error values at the edges of design space are high [71].

2.3.3.3 Space-Filling DoEs for Asymmetric Design Spaces

Space-filling experimental designs can be modified for constrained problems (i.e. with asymmetric design spaces) by removing the infeasible points from the original design [75]. However, the final design might not result in a high fidelity model due to either poor properties of final design or lack of feasible test points to cover all design space. Therefore, several DoE techniques have been proposed to generate space-filling designs for constrained design spaces [76], [96], [99]. Although these methods have not been employed for engine applications, some of these methods are reviewed in this section to study the strategies used by these methods to handle design constraints.

Draguljic *et al*. [96] proposed the CoNcaD (Constrained Non-collapsing Design) algorithm as a space-filling DoE technique for high dimensional problems with nonrectangular design spaces. The CoNcaD technique starts with selecting a random initial solution set and gridding the design space, then modifying the grids to have points at the feasible parts of nonrectangular design space (due to constraints). This process is followed by selecting the *N* best solutions among the candidates, after calculating the modified Audze and Eglais function [96]. This function is based on maximising both space-filling and projected distances. However, the main drawbacks of this strategy are:

1) The requirements of selecting different arbitrary numbers to initiate the design, such as Q (Q>N) where Q is an arbitrary number bigger than the total number of samples (N). This term (Q) defines the number of initial candidates, from which N samples with better performance will be selected (based on space-filling and projected distances). So, if running the simulator for each sample is expensive, this process is inefficient. Also, if not enough sample points from the candidates are feasible, the DoE should be redesigned.

2) Considering the effects of initial solution set on the quality of final design, the algorithm should be run several times from random starts, to choose the best design as the final experiment set.

3) Increasing the number of initial candidates to enhance the final design properties has a drastic effect on the computation efficiency of the strategy.

Stinstra *et al*. [76] proposed a space-filling design for complex problems with nonrectangular design spaces. This optimisation-based strategy distributes the points within the design space one at a time using the Maximin criteria. Distribution of sample points starts from an initial random point, and then the next samples are selected subsequently after checking the design constraints to augment the overall design space-fillingness. A standard NLP (non-linear programming) solver was employed to solve this optimisation problem. The main shortcomings of this strategy are:

1) This strategy does not guarantee good projective properties since the quality of design can change based on the selected initial sample solution.

2) Maximin criteria tends to start the sample generation around the corners , especially for high dimensional problems [96].

3) The computation efficiency drops drastically when the number of variables increases.

4) For large number of samples, the NLP solver capacity can be a problem.

2.3.4 Shortcomings of Single-Level DoE Strategies

Single level DoE strategies are commonly used to collect all the required data in advance to the model fitting stage. However, one-shot strategies are not always the best DoE option, particularly when the system behaviour is unknown [100], [101]. Applying single-level DoEs for unknown systems raises the risk of test plans that a larger design is selected due to the possibility of *over-sampling*, thus wasting time and energy by collecting more tests than needed [101]. Conversely, test plan might generate an insufficient amount of information due to *under-sampling*, with the implication that the required model accuracy is not achieved. In this situation, there is a need to loop back and collect more data,

as illustrated in Figure 2.9. The main reasons for necessity of loop back iterations can be summarised as [57]:

1) Infeasible test points due to design constraints or range limits of the parameters.
2) Poor model quality due to excessive measurement variation, outliers, or choice of model type.
3) Non-observation of a design region, since not enough data is collected from regions of interest.

All these reasons influence the model quality eventually. It is noteworthy that the importance of the above reasons can be compromised for different types of DoE strategies. For instance, space-filling designs are generally less sensitive to any changes in the placement of test points [102], however; the design properties of classical and optimal designs can be affected severely.

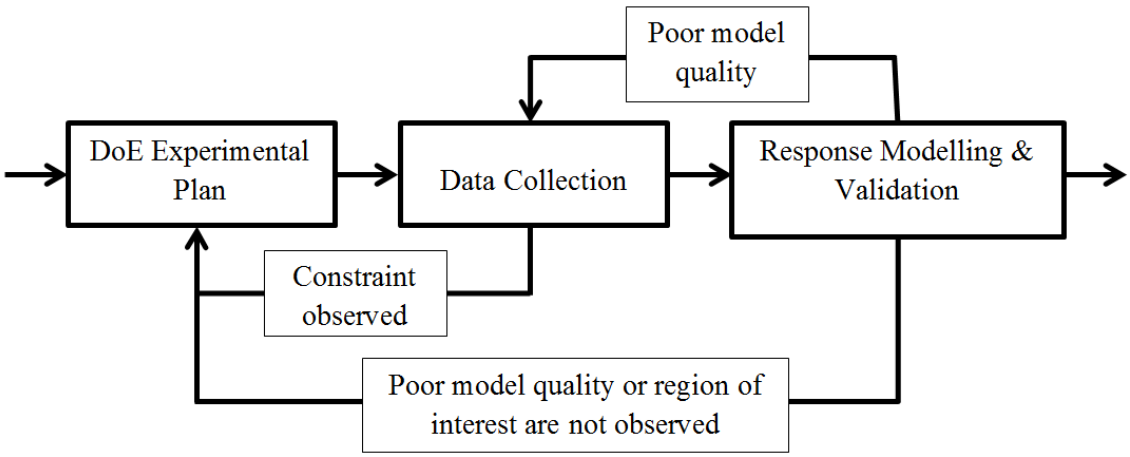


Figure 2.9: Loop back iteration to improve a DoE experimental plan

Recent research work in fields dealing with testing cost issues (e.g. electronics, chemistry and aerodynamics) focused on the development of sequential DoE approaches, which are also known as adaptive sampling [103] or active learning [104]. The idea of sequential DoE approaches is to iteratively augment an initial DoE with further test points until the desired model quality is reached [96], [100], [105], [106]. This strategy can facilitate a higher testing efficiency compared to the fixed size tests commonly used in practice, and has the advantage that it can flexibly adapt to modelling complexity requirements of various problems.

Considering the importance of selecting an efficient experimental design to reduce the testing costs [107], the existing sequential DoE approaches will be reviewed in the following section.

2.3.5 Sequential DoE Strategies (adaptive DoEs)

In general, sequential DoE strategies start with an initial experimental design, followed by an iterative augmentation process. This iterative process continues until the desired model fidelity is achieved [65], [100], [108], [109].

2.3.5.1 Classification of Sequential DoEs

Sequential DoEs can be divided into two main categories:

1) Optimal sequential design: For this type of sequential DoEs, the model type and its parameters are known in advance (e.g. polynomials). This allows the algorithms to use the behaviour of the set model type to guide the sampling points into the right direction within design space, e.g. the D-optimal designs minimise the covariance of the model parameters estimates [110], [111]. The main issue with these DoEs is that if the assumed model type is not suitable for the response, the DoE plan is not efficient and the enhancement in model accuracy through collecting more data is not guaranteed.

2) Evolutionary sequential design: Given that the type of model may not be known in advance for many engineering problems and therefore a nonparametric model is required, justifies the need for a generic sequential DoE which makes no assumptions about the model type, number of sample points or system behaviour. Such DoEs use the information from the previous iterations to decide where to select the next test point [100].

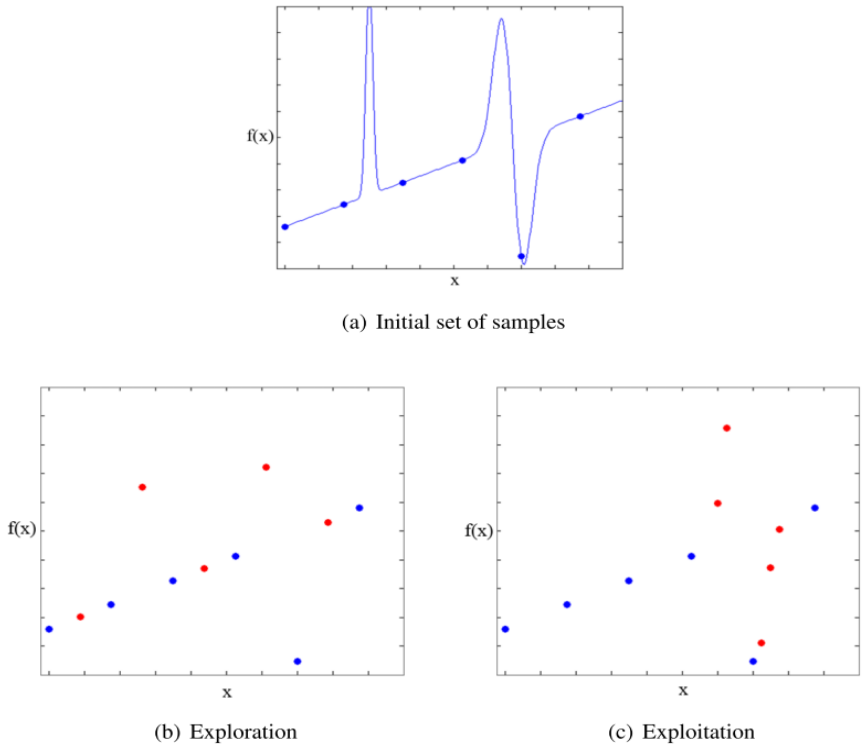
In general, the evolutionary sequential DoEs can be further classified into:

(i) Exploration-based sequential designs [65], [106], [108], [109]: Exploration-based sequential DoEs, give equal importance to all regions of design space and aim to fill it up as evenly as possible at each sequence. In this method, the location of the test points from the previous iteration is used as feedback for sampling new test point, ensuring that not too many or too few samples are collected from the same regions of design space. These DoEs

are not specifically linked with any response models and aim to distribute the points evenly through the design space.

(ii) Exploitation-based sequential designs [105], [108]: Exploitation-based sequential DoEs use an error measure from the previous steps to guide the sampling points to the interesting parts of design space, e.g. areas with discontinuous system behaviour or areas containing optima. The main problem with exploitation-based DoEs is the tendency to over-focus on specific areas, which could leave some part of design space undersampled. This problem may result in early dismissal of potential optimum solutions given that some parts of the high dimensional design space could be left unexplored.

For the evolutionary sequential DoEs, the main consideration for the augmentation designs is the trade-off between exploration and exploitation [65]. The importance of the trade-off between exploration and exploitation is illustrated in Figure 2.10. In Figure 2.10 (a), the response function and initial experimental design are shown using blue lines and dots, respectively. For the hypothetical case that the response function is unknown, to predict the response behaviour using the sample points, it seems that the function behaves linearly except at a single sample point to the right. Accordingly, to collect more information about the function, the initial DoE is augmented once by an exploration-based algorithm, Figure 2.10 (b), and once by an exploitation-based algorithm, Figure 2.10 (c). The exploration-based algorithm distributed the additional points within the design space evenly regardless of the nonlinearity at the right, which has resulted in exploration of a new nonlinearity in the design space. On the other hand, the exploitation-based algorithm missed the second nonlinearity (i.e. located to the left) by focusing on collecting more data around the first nonlinearity area. Thus, it can be concluded that sequential experimental designs must consider exploration to a certain degree [65].



(a) Initial set of samples

(b) Exploration

(c) Exploitation

Figure 2.10: Trade-off between exploration and exploitation [108]

It should be noted that for exploration-based sequential DoEs, the same criteria as space-filling designs (i.e. space-fillingness, projection properties and orthogonality) are used to compare efficiency of different strategies. In addition to these criteria, granularity is also an important requirement for sequential designs [65]. Granularity indicates the proficiency of the DoE strategy to augment the initial experimental design by small batches of additional test points (preferably 1 test point). Accordingly, a fine-grained sequential strategy is flexible regarding the total size of DoE samples, despite the number of design variables and levels, which consequently results in avoiding over- or under-sampling [54], [101].

In general, considering the fact that the sequential approaches are implemented with less prior information about the system (e.g. the total number of sample points or the model type), it is expected that the final sequential design perform worse (e.g. less computation efficiency, and design properties) than pre-optimised one-shot designs. However, if the total number of required sample tests to achieve a satisfactory model accuracy is reduced, the overall efficiency of the strategy is superior by avoiding over- or under-sampling.

2.3.5.2 Exploration-Based Sequential Design

Random sampling [108] is the simplest sequential DoE method in which the additional test points are selected randomly without considering the location of previous test points. This cheap approach might perform satisfactory for large set of sample points, but for small batches of test points both space-filling and projective properties are expected to be deviated from an optimum DoE properties [108].

Some authors [112], [65] proposed to use the Monte Carlo strategy, which is established on the concept of repeated random sampling. For these designs, Monte Carlo strategy is employed to generate a large batch of random candidates within the design space, and then rank each of the candidates based on an optimality criterion. This criterion is the aggregate of intersite and projected distances with a coefficient, scaled between [0 1], to control the focus of ranking objective towards maximising any of these two distances.

Low-discrepancy DoE strategies, also called quasi-random sequences or quasi-Monte Carlo [113], are similar to random sampling. In this strategy, DoE is generated through sequences of sample points (*N* sample points) where the discrepancy of any taken subset (with *P* sample points where $P < N$) is low. A subset has a low discrepancy when the number of selected sample points within an arbitrary subpart of design space is 'close to proportional to a particular measure of size for this subset' [65], therefore; the discrepancy can be a variant term regarding the subpart design shape and the set measure size. Low-discrepancy DoE methods can satisfy the projection requirement, however; the space-filling property is usually unsatisfactory. Furthermore, not all the low-discrepancy methods are an efficient sequential sampling technique. For example the Hammersley sequence [114] requires the total number of final DoE before generating the sample points. The most well-known sequential methods using low-discrepancy strategy are Sobol sequence [115] and Halton sequence [114] which do not require prior knowledge regarding the total sample points.

To enhance the space-filling property of sequential experimental methods, several non-optimisation based algorithms were developed, such as the Voronoi-based Sequential Design and Delaunay-based Sequential Design

[108]. The simplest way to perceive the groundwork of Voronoi-based DoE method is to study the Voronoi tessellation (or Voronoi diagram), shown in Figure 2.11. This figure illustrates a number of Voronoi cells which are distinguishable using a colour scheme. A Voronoi cell includes all the sample points closer to a fixed site, which is expressed as the area trapped between intersections of the Voronoi edges. A Voronoi edge indicates the boundary where the closest neighbour changes [116]. In Figure 2.11, larger Voronoi cells are shown by a darker colour, such as the cells around the corner. Therefore, to achieve a final experimental design with better space-fillingness property, the additional samples should be distributed in larger cells. In this method, Monte Carlo approach is used to estimate the volume of each Voronoi cell, however; this process is time-consuming for high dimensional problems (more than 6 variables). This sequential space-filling DoE strategy can preserve a good intersite distances [108], whereas it cannot guarantee a good projected distance. Also, it is not expected to see a strong correlation among the columns of variables due to random selection of points inside the Voronoi cells.

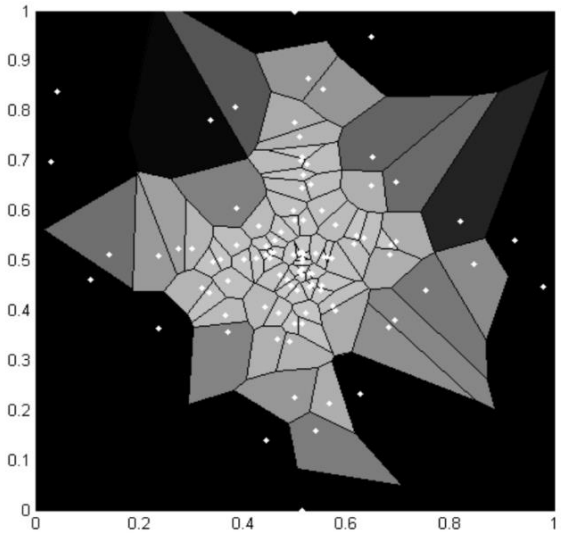


Figure 2.11: A Voronoi diagram [108]

Delaunay triangulation is the dual of the Voroni diagram, illustrated Figure 2.11. Therefore, Delaunay-based sequential DoE approach [108] is based on the same concept as the Voronoi diagram, thus sharing similar deficiencies. It should also be added that the process of estimating the volume of a Delaunay triangulation is comparatively more complex than estimating the volume of a

Voronoi cell, and consequently more time consuming. So, implementation of a Voronoi-based DoE is usually preferred to a Delaunay-based DoE [108].

Sequential nested Latin Hypercube strategy [117], [118] is also a non-optimisation based sequential design with the idea of preserving the system structure similar to the LH designs, and then selecting the additional test points by refining the existing LH grids. This structure delivers a good projected distance, due to stratified structure of LH designs, whereas good intersite distance demand is satisfied through finding the local optimum location of the additional test points one at a time. For this strategy, with *N* initial sample points after *p* iterations, the delivered DoE when $N + (N-1)(2^p - 1)$ samples are selected is an exact Latin Hypercube design. Therefore, if the total number of DoE sample points can be predicted in advance, with a proper guess of initial test size *N*, a suitable final design can be delivered after *p* iterations. The main pitfall of this design is poor granularity. For instance, the final size of the LH in the proposed nested LH strategy by Qian [117] can only be a multiple of its subset sample size, and also the nested LH designed by Husslage [118] can only contain two layers of nested LH designs.

Another well-known attempt to adopt the LH designs to a sequential strategy is quasi-Latin Hypercube [77]. This method is based on the flexibility to locate the LH sample points not exactly on the allocated grids. In this design, a parameter is introduced to define the minimum possible distance between the projected points on one of the axes. If this parameter is close to 1, this design is a traditional LH, and if the parameter is close to 0, the design is an unconstraint space-filling design (using Maximin criteria). However, the space-filling quality and projective property of the design can be affected progressively for complex engineering problems (with many design variables) after several sequences of design augmentation [65].

In most of the above sequential approaches [108], [114], [115], location of the new test points is determined through a non-optimisation based process, in a way to deliver either a good intersite or projective distance (or both) over the total design. However, an alternative idea is to augment the designs iteratively using an optimisation-based approach [119]. It should be highlighted that

selecting the optimum location for the additional test points is a complex optimisation problem itself [65]. For instance, Figure 2.12 illustrates the optimisation surfaces for a 2-dimensional problem, produced by (a) intersite distance, and (b) projection distance. This figure shows the location of 20 fixed sample points, and potential optimum locations for selecting the 21st test point associated with intersite or projection distances, as shown in Figure 2.12 (a) and (b). It can be seen that for both objectives there are a considerable number of local optima exists, whereas only one of them is the global optima. The optimisation problem gets even more erratic when optimisation objective is to maximise both distances, as shown in Figure 2.12 (c).

Xiong *et al*. [119] proposed an exploration experimental strategy based on using a global optimisation algorithm, i.e. Genetic Algorithm (GA), to provide a design with good intersite and projected distances. In this method, the initial design is principled on LH sampling technique, so it is also called sequential Quasi-LHD. The objective of this optimisation problem is to distribute the points within the design space based on Maximin criterion [80], subject to the defined constraint for the minimum allowable projection distance onto subspace of each parameter. This design can provide a DoE with good statistical properties for high dimensional problems (e.g. space-filling, projection, orthogonality and granularity), however; it can be directly influenced by the trade-off balance between the space-filling and projection requirements. Also, this process is based on GA which is not always reliable for solving constrained optimisation problems. It can be argued that GA algorithm is comparatively time-consuming in dealing with constrained optimisation problems [43].

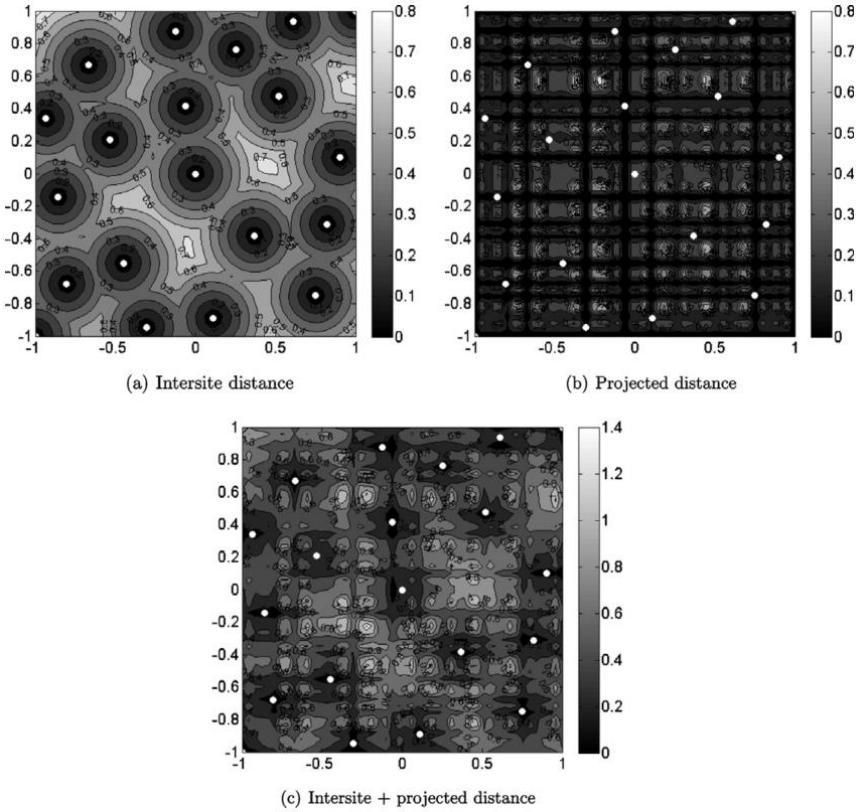


Figure 2.12: The optimisation surfaces of intersite distance, projection distance, and combination of both distances, for a design of 20 points in 2D design space [65]

2.3.5.3 Exploitation-Based Sequential Design

Different types of exploitation-based sequential DoEs have been introduced in literature, such as LOcal Liner Approximations (LOLA) [100] and expected improvement [17] methods. However, all sequential designs require space-fillingness (or exploration) to a certain degree to avoid dismissing an optimum solution, as illustrated in Figure 2.10 [65].

Accordingly, different types of hybrid sequential DoEs have been developed to consider both exploration and exploitation demands. For instance, Crombecq *et al*. [100] developed a hybrid sequential DoE, which integrates both exploration (Voronoi tessellation [108]) and exploitation (LOLA [100]) strategies. LOLA is grounded on the idea of ranking samples through the local linearity of the function. Therefore, by applying this method, fewer points are selected where the function is linear, and in effect is easier to predict the behaviour than the regions with large variance. In this strategy, the exploration part is based on a Voronoi diagram, which is not an efficient algorithm for high dimensional

problems. The main shortcoming of this hybrid strategy is the deficiency of delivering a final design with a good projected distance.

2.4 Response Surface Modelling

Response surface modelling (RSM) is a collection of statistical and mathematical techniques to define the empirical relationship (f) between independent input parameters (X) and responses of interest (Y) [19]. The relationship between input parameters and responses can be shown in a general form as [56]:

$$Y = f(X) \qquad \text{Equation 2.6}$$

Response surface modelling was originally developed to model experimental responses, however; it has also been used frequently for numerical experiments [18]. Some of the main applications of RSM in engineering design are as following [17]–[19]:

- RSM can be used as a relatively cheap alternative to expensive analysis methods, such as finite element methods.
- RSM can be used to analyse the effect of each independent variable on the responses behaviour.
- RSM can be used as a filtering system, given that RSM empowers handling noisy data by using various approximation methods.

Considering that the structure of relationship between the parameters and responses are unknown for most of the real-life problems, selecting the best statistical technique to approximate the true relationship is a challenging task [72]. Generally, various response surface modelling techniques can be divided into two main categories [55]:

I. **Parametric models:** parametric models are explicitly dependent on the underlying model structure, thus demanding prior knowledge regarding the response behaviour [19]. For these models, the unknown model parameters for the pre-defined model type are estimated based on the experimental measurements (i.e. test points). In the following section,

polynomials [80], [120], which are well-known parametric models, are reviewed in details. Polynomials have been commonly used as modelling technique for engine model based calibration [55], [56].

II. **Non-Parametric models:** non-parametric models do not require explicit model assumptions [55]. Therefore, these models are an attractive modelling option for the engineering cases when no prior knowledge regarding the suitable model type is available. Given that non-parametric models have recently gained many attractions in the area of engine model based calibration [55], [56], two of the mostly used non-parametric modelling techniques in model based calibration [22], [24], [121] are reviewed in this section:

- Gaussian Kriging model [17], [122].
- Neural network models [123], in particular Radial Basis Function (RBF) model [49].

2.4.1 Polynomial Models

Polynomial models predict a response behaviour (y) based on the input parameters (*X*) by linear combination of a set of base functions (f), as illustrated in Equation 2.7 [80], [120].

$$y(x) = f^{\mathrm{T}}.a = \sum_{p=1}^{P} a_p.f_p(X) \qquad \text{Equation 2.7}$$

P indicates the number of base functions, and a is the parameter vector. The parameter vector is retrieved through a linear regression based on least squares estimation of sum of squared errors (SSE) [55], as shown in Equation 2.8. In Equation 2.8, the model prediction error is defined as the discrepancy between the estimated response value for a measured sample point ($\hat{y}$) and the real response value (y).

$$SSE = \sum_i (y_i - \hat{y}_i(a, x_i))^2 \qquad \text{Equation 2.8}$$

The main reasons for popularity of polynomials can be summarised as:

- Polynomials have a simple structure, so they are easy to understand [56].

- Polynomials requires low computational effort due to linearity in the unknown parameters, which consequently led to a fast modelling process [55].
- Polynomials show less possibility of over-fitting, which is particularly important for modelling based on noisy measurements [49].

Regardless of these advantages, there are some limitations in using polynomial models. Some of these limitations are:

- Polynomials require prior knowledge regarding the response behavior. Therefore, if the model type is not selected properly, the required model accuracy might not be delivered [80], [120].
- For complex responses, where a higher order of polynomials is required, the number of model's parameters increases significantly. Accordingly, more experimental measurement would be required to calculate the model's parameters [49], which leads into a time consuming and expensive modelling process.

2.4.2 Gaussian Kriging Model

Gaussian Kriging model is a non-parametric modelling technique, in a sense that this approach determines the most likely underlying parameters for a set of measured sample points, instead of determining a fixed set of parameters for a specified model type (like the Polynomials) [55]. The theoretical basis of Gaussian Kriging models was developed by Matheron [124], based on the original work by Danie G. Krige, a South African geologist who studied mining data [17].

Gaussian Kriging model is derived from Equation 2.9 [75], [125], which is combination of a global model (f), and localised departures (G).

$$y(x) = f(x) + G(x) \qquad \text{Equation 2.9}$$

In this equation, the response y is predicted at a point x using a global approximation function f, which is usually a polynomial, and 'localized' deviations G, which are calculated by interpolation of the measured sample

points. $G(x)$ denotes the realization of a weakly stationary stochastic process with mean 0, process variance $\sigma^{2,}$, and nonzero covariance function, as given by Equation 2.10 [49]:

$$Cov\left(G\left(x^i\right), G\left(x^j\right)\right) = \sigma^2 R(x^i, x^j) \qquad \text{Equation 2.10}$$

$R(x^i, x^j)$ represents the correlation between any two of the measured sample points, x^i and x^j. *R* is assumed to be a function of small set of parameters, which are estimated based on the Likelihood function [17]. Likelihood function defines the probability of measured sample points [126].

Some of the main advantages of Kriging models in comparison with polynomials are [22], [49], [127]:

- Kriging models are more flexible due to wide range of the correlation functions.
- Kriging models show great accuracy in building global approximation models.
- Kriging models require less data collection due to strong interpolation among the measured sample points.
- Kriging models can either 'honour the data' by providing an exact interpolation of the data, or 'smooth the data' by providing an inexact interpolation.

A potential pitfall of using Gaussian Kriging models is the *curse of over-fitting* [17], particularly for a noisy set of measurements. Also, implementation of Kriging models is relatively a time consuming process since determining the maximum likelihood parameters is a complex optimisation problem [49].

It is noteworthy that performance of Kriging modelling technique can be endangered by the type of DoE used to collect the experimental data. This method has shown some difficulty in fitting response models for full factorial and central composite DoE methods [128], since the correlation matrix becomes singular when multiple sample points are located close to each other.

2.4.3 Neural Network (NN) Methods

Artificial neural networks, which are known as Neural Network (NN) [55], are non-parametric computational models which are inspired by animal's nervous system (in particular brain performance) [49]. Human brain computes in an extremely nonlinear and parallel manner which is completely different from a conventional digital computer. Biological neural system is based on small units called 'neurons' which are strongly connected to each other and have the ability to process certain computations much faster than supercomputers [49].

Neural Networks consists of several layers [129]:

- Input layer: the first modelling layer which is not cooperating in computation process.
- Output layer: the last modelling layer which produces the output.
- Hidden layers: the layers between 'Input' and 'output' layers. Each 'Hidden' layer may consist of several neurons itself, which are only connected to the adjacent layer.

Neural Network layers participate in the model fitting process through the following steps [129]:

1- The measured data is fed to the modelling system through the 'Input' layer.

2- The data is propagated into the 'Output' layer using several 'Hidden' layers, where each 'Hidden' layer has a weight and activation function itself [123].

3- The difference between the approximated response from the 'Output' layer ($\hat{y}$) and the measured response (y) is calculated using Mean Squared Error (MSE) [49], as illustrated in Equation 2.11, to learn the relationship between the input parameters and output. Subsequently, the layers' weights are modified and propagated back to modify the calculation, this process is called 'Training'. System learning process can be executed using one of these three methods: supervised, unsupervised or reinforcement. The most commonly used learning type is the supervised back-propagation training algorithm [129].

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \qquad \text{Equation 2.11}$$

Radial Basis Function model was introduced as a variant of NN models in late 1980s [130]. A general form of a RBF model can be mathematically described by the following equation [49]:

$$f(x^i) = \sum_{i=1}^{n} w_i \phi(\|x^i - x^j\|) = t^i \qquad \text{where} \quad i \neq j \qquad \text{Equation 2.12}$$

Where $f(x^i)$ is the RBF function, *n* is the number of measured points, $\phi(.)$ indicates the basis function, w_i denotes the weight of basis functions, t^i is the measured output, and $\|x^i - x^j\|$ is the distance between the measured points *i* and *j*. The interpolation properties of RBF models can be changed based on the basis function type, such as Gaussian functions, Multi-Quadratic functions, Thin Plate Spline functions, Cubic functions, and Linear functions [49].

Gaussian function is the most commonly used basis function [55] for RBF models, as given in Equation 2.13. This function is working based on a set of centre points (x), which are a subset of measured data, and radial distance, which determines the discrepancy between the centre point and the model prediction (x^*). A radius vector is also used to adapt the impact of basis functions (*r*).

$$\phi(x_i{}^* - x_i) = exp\left(-\frac{|x_i{}^* - x_i|^2}{r_i^2}\right) \qquad \text{Equation 2.13}$$

The main advantages of Neural Network computational models are:

- The ability to learn the complex and nonlinear relationship between the input and output parameters [123].
- Confronting the missing and noisy data with a good generalization capability [123].
- NN requires fewer measurements than other modelling methods. It is due to the fact that NN models are purely driven from the collected data through a learning (instead of assuming a model type like the Polynomials) [131].
- NN is very fast in learning the relationship between the input and output variables because of using a two stage network training. The first stage is to determine the weights from the 'Input' to the 'Hidden' layer, and the second

stage is to determine the weights from the 'Hidden' layer to the 'Output' layer [49].

Although Neural Network models have shown to be an efficient modelling technique for complex problems, such as modelling of internal combustion engines [132], there is always the possibility of over-fitting [17], particularly for noisy measurements.

2.4.4 Model Validation

The main objective of model validation is to investigate how accurate a fitted statistical model can predict true behaviour of a response. There are different methods to validate the accuracy of the fitted model:

- **Physical behaviour:** the first method is to validate the response behaviour based on the physical interactions of the input variables [133]. This method of validation can ensure that a statistical model is not over-fitting, however; it requires prior knowledge about the response behaviour.

- **Internal validation:** internal validation technique is based on investigating model's statistical properties using different statistical methods, such as Root Mean Square Error (RMSE) [56], as given by Equation 2.14. RMSE is principally calculated by the discrepancy between the real value (y) of the measured sample points (n) and the corresponding prediction values ($\hat{y}$).

$$RMSE = \sqrt{\frac{1}{n} . \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \qquad \text{Equation 2.14}$$

  PRESS RMSE (Prediction Error Sum of Squares) [54] is also an internal validation criterion, which is a useful cross-validation technique for investigating over fitting [71], [134]. PRESS RMSE is calculated by fitting the statistical model to 'n-1' of the measurements, and predicting the response value for the remaining sample point. The difference between the actual value and the predicted value of the remaining sample point is called prediction residual, and sum of the squares of all the predicted residuals is PRESS [134].

- **External validation:** external validation technique requires an additional set of measurements (e.g. *v* sample points) to be used for validation of model's predictive performance [48]. These measurements are not used for fitting the response surface. Accordingly, the fitted response model to *n* measurements is used to calculate the response values of the *v* validation sample points. There are different external validation criteria, such as Validation RMSE [56], given by Equation 2.15, and Relative Error [55], illustrated by Equation 2.16, that can be used to investigate a model's accuracy. These validation criteria are based on the discrepancy between the predicted values and the measured values of the validation set.

$$Validation\ RMSE = \sqrt{\frac{1}{v} . \sum_{i=1}^{v} (y_i - \hat{y}_i)^2} \qquad \text{Equation 2.15}$$

$$Relative\ Error\ (\%) = \sqrt{\frac{\sum_{i=1}^{v} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{v} {y_i}^2}} \qquad \text{Equation 2.16}$$

2.5 Review of DoE Strategies and Statistical Models Used for Engine Mapping

Several studies are available which have investigated performance of various modelling techniques in relation to some of the frequently used DoE types, and provided some guidelines and recommendations [87], [135]. Some of these recommendations are summarised in Table 2.4.

Experimental Design	**Model Choice**
Fractional Factorial	Polynomial (Linear, Quadratic)
Central Composite	Spline (Linear, Cubic)
Box Behnken	Multivariable Adaptive
Optimal	Regression Spline
Latin Hypercube	Artificial Neural Network
Sequential Methods	RBF
Hybrid Methods	Kriging

Table 2.4: Recommendations for DoE and model types

In engine model based calibration context, both Design of Experiment and modelling technique are vital elements to deliver highly accurate engine models. Given that most of the calibration engineers are not educated in mathematics, they prefer to use a modelling technique which is [55]:

- Easy to understand, without having background knowledge about the model's algorithm.
- Easy to interpret based on the physical insight, by inspecting the model's behaviour against the input parameters.

Therefore, Polynomials are the most frequently used modelling techniques for engine model based calibration [1]–[6], since they are easy to use and interpret [56]. By using Polynomials, calibration engineers can easily select a suitable order of polynomial based on their knowledge about the underlying model structure (unlike the non-parametric modelling techniques), and also can estimate the model parameters in terms of dependency of an output on the input parameters. Moreover, due to linearity of the parameters, various statistical metrics are available to validate the model behaviour, e.g. defining potential outliers, confidence intervals for the model prediction, and etc. [55].

Regardless of the available guidelines in literature, as illustrated in Table 2.4, both optimal and classical DoE designs have been frequently used by calibration engineers to fit Polynomial models [1]–[6].

Non-parametric modelling techniques, such as RBF and Kriging models, have also been used by calibration engineers as an attractive alternative for Polynomials [22], [24], [58], [136]. Kriging models perform relatively robust in handling outliers, compared to the RBF models [135]. Also, interpretation of Kriging models is easier since they provide information about the underlying model structure, whereas RBF models are black boxes with a little possibility of model interpretation [55]. RBF models are mainly interpreted using graphical inspections [121]. For using non-parametric modelling techniques in engine model based calibration, space-filling DoEs and hybrid designs (i.e. combination of two or more designs) have usually been employed [22], [24], [58], [136].

Table 2.5 and Table 2.6 summarise some examples of used DoE methods and modelling techniques in engine model based calibration (for both gasoline and diesel engines).

Reference	Purpose/Parameters No	DoE Type	Points No	Fitted Model
Ryu *et al.* [58]	Optimisation of catalyst light-off time during cold start (7 parameters)	D-Optimal + LH	100	Quadratic-RBF
Schlober *et al.* [1]	Map optimisation (4 parameters)	V-Optimal + Space-filling	99	Cubic
Singh *et al.* [2]	Optimisation of camshaft control calibration (2 parameters)	V-Optimal	24	Cubic
Steidten *et al.* [3]	Gas exchange model calibration (8 parameters)	D-Optimal	112	Cubic
Lach *et al.* [4]	Application of DoE for valve train development (8 parameters)	Central Composite	82	Cubic
Seabrook *et al.* [22]	Calibration of gasoline engine with direct injection and variable valve timing (5 parameters)	OLH	52	Kriging
Morton *et al.* [24]	Optimisation of a dual dependant variable valve-timing engine calibration (5 parameters)	Space-filling	96	RBF

Table 2.5: References of DoEs applied for Gasoline engine

Reference	Purpose/Parameters No	DoE Type	Points No	Fitted Model
Debavelaere *et al.* [136]	Optimisation of the Diesel combustion system (3 parameters)	Central Composite + LH	40	Kriging
Chang *et al.* [5]	Minimizing emission deviation due to injectors (5 parameters)	Online D-Optimal	-----	Cubic
Lach *et al.* [6]	Application of DoE for base engine design (8 parameters in 3 levels)	Central Composite	82	Cubic

Table 2.6: References of applied DoEs for Diesel engine

2.6 Summary

The steady state engine mapping and calibration process is discussed in this chapter, with a focus on different DoE methods and statistical modelling techniques. The DoE methods are reviewed within two main categories of single-level and sequential strategies. The single-level DoE strategies have been broadly applied for engine applications, such as standard designs, optimal designs and space-filing designs. Although both standard and optimal designs have been frequently used for engine mapping purposes, the initial requirements of pre-defining the model type, number of test points, and the feasible design space affect the practicality of these designs for unknown engine applications. Thus, space-filling DoE techniques, such as OLH designs,

have been frequently used recently by engine calibrators due to their flexibility regarding the model type and design space. Space filling designs are not fitted based on any model assumptions and only tries to cover the whole design space uniformly, however; it still requires the total number of required test points in order to avoid over / or under sampling. Different types of sequential DoE strategies are also analysed in this chapter. These strategies can provide the opportunity to avoid either of over / or under sampling by sequential modification of an initial DoE. However, the applied modification technique (e.g. exploration-based and exploitation based strategies) has a substantial effect on the efficiency of sequential DoE strategies (i.e. to achieve a response model of target accuracy with less test points).

Afterwards, some of the existing modelling techniques, both parametric (e.g. Polynomials) and non-parametric (e.g. RBFs and Kriging), are discussed. Generally, non-parametric models have the potential to predict the behaviour of highly complex systems, however; there is always the danger of over-fitting.

Chapter 3: Review of Optimisation Methods and Their Application to Engine Calibration

3.1 Introduction

In this chapter, the existing literature on optimisation frameworks is reviewed in three main categories: 'single objective optimisation', 'multi-objective optimisation', and 'multidisciplinary design optimisation' methods. This review includes explanation of each framework's principles, followed by comparing special properties of algorithms at each category. Thereafter, the focus turns to review the available literature on application of different optimisation frameworks for engine calibration optimisation problems, particularly gasoline engines.

3.2 Optimisation Problem

Optimisation problems aim to find the optimum design vector *X* that maximises (or minimises) the objective function *f(X)* subject to the equality *h(X)* and inequality *g(X)* constraints [137]–[139]. An optimisation problem can be stated in a general form as given in Equation 3.1.

$$\begin{aligned} &\textit{Minimise / Maximise} && f_m(X) && m = 1,2,\dots,M \\ &\text{with respect to} && X && \\ &\text{subject to} && g_j(X) \le 0 && j = 1,2,\dots,J \\ & && h_k(X) = 0 && k = 1,2,\dots,K \\ & && LB_i \le x_i \le UB_i && i = 1,2,\dots,n \end{aligned} \qquad \text{Equation 3.1}$$

In this equation, *LB* and *UB* define the boundary limits for the design variable *x*.

If the optimisation problem involves one objective (*M = 1*), the problem is a single objective optimisation (SOO) problem. However, sometimes the engineering problems require optimisation of several objectives simultaneously (*M > 1*) where some objectives might have conflicting nature [137], [140]. These optimisation problems are called multi-objective optimisation (MOO) problems.

The idea of optimisation algorithms is to search for a design vector which delivers the optimum objective value for an optimisation problem [141]. Many optimisation algorithms have been developed to solve complex optimisation problems, particularly since the growth in computing hardware development. In this section, some of the main algorithm developments for solving single objective and multi-objective optimisation problems are reviewed and their performances are compared together.

3.3 Single-Objective Optimisation (SOO) Algorithms

Many optimisation algorithms have been developed to solve single objective optimisation problems. Generally SOO algorithms can be divided into two main categories [140]: 'classical methods' and 'evolutionary methods'.

3.3.1 Classical Methods

Classical SOO algorithms can also be reviewed in two primary categories based on the searching techniques they use: 'gradient based' algorithms [139], [142], [143], and 'direct search algorithms' [144].

3.3.1.1 Gradient-Based Algorithms

In gradient based algorithms, the searching process is working based on the functional gradient information [145]. Sequential Quadratic Programming (SQP) algorithm [146] is one of the well-known gradient based algorithms which has been frequently used as a powerful tool for solving constrained non-linear optimisation problems.

Various techniques have been proposed for handling the constraints of gradient based algorithms, such as Lagrange Multipliers (LM) [145] and Kuhn-Tucker Conditions (KTC) [146]. Lagrange Multipliers (LM) technique is mainly used for optimisation problems with equality constraints. In this technique, firstly the Lagrange function is created as shown in Equation 3.2 [146].

$$L = f(X) + \lambda_j(c_j - g_j(X)) \qquad j = 1,2,\dots,J$$

with respect to $X = [x_1, x_2, \dots, x_n]$ Equation 3.2

where X is the vector of n design parameters x_i, L indicates the Lagrange function, $f(X)$ is the objective function, $g_j(X)$ is one of the J constraints, c_j is the limit for constraint j, and λ_j is the multiplier factor corresponding to constraint j.

Afterwards, the partial differential of Lagrange function with respect to each of the design variables is set equal to zero, as given by Equation 3.3. In this way, the constrained optimisation problem is converted to a system of $n+J$ equations with $n+J$ variables.

$$\partial L/\partial x_i = 0 \qquad i = 1,2,\dots,n$$
$$\partial L/\partial \lambda_j = 0 \qquad j = 1,2,\dots,J \qquad \text{Equation 3.3}$$

Kuhn-Tucker Conditions (KTC) is an efficient technique for handling both equality and inequality constraints [137], [142] (i.e. unlike LM technique which only handles equality constraints [145]). In this technique, which is also based on the Lagrange function given in Equation 3.2, KTC conditions are necessary (but not enough) to be satisfied if a solution is an optimum. These conditions are given in Equation 3.4. Accordingly, to guarantee a KTC solution is an optimum, some additional conditions are demanded. These conditions are called Kuhn-Tucker sufficiency conditions. Based on this conditions, if a solution satisfies all the KTC conditions while all the objective and constraint functions are differentiable and concave in the non-negative part of design space, the solution is an optimum solution [146].

$$\partial L/\partial x_i \le 0 \qquad x_i \ge 0 \qquad i = 1,2,\dots,n$$
$$x_i \times \partial L/\partial x_i = 0 \qquad i = 1,2,\dots,n$$
$$g_j(X) \le c_j \qquad \lambda_j \ge 0 \qquad j = 1,2,\dots,J$$
$$\lambda_j \left(c_j - g_j(X)\right) = 0 \qquad j = 1,2,\dots,J \qquad \text{Equation 3.4}$$

Gradient based algorithms have shown to converge very quickly [146], which means they are computationally efficient. However, considering that the searching process of these algorithms is based on gradient information, this type of SOO algorithms cannot be used for problems where objective function or constraints are not differentiable [146]. Also, gradient based algorithms might not be an efficient approach for solving complex optimisation problems with

multiple optima, given that there is a great chance that these algorithms converge to a local optimum [137], [147].

3.3.1.2 Direct Search Algorithms

Searching process of direct search algorithms is based on a sequential examination of the trial solutions, which includes comparing the current solution with the best solution up to that time, and determining the next trial solution based on the earlier examinations [148]. Pattern search algorithm is one of the key algorithms which has shown a robust convergence behaviour using a direct search technique [144], [148].

Given that no gradient information is required for direct search methods, these algorithms are relatively easier to be implemented. Also, direct search algorithms can be applied for a range of complex optimisation problems where objective or constraint functions are non-differentiable. The main shortcoming of direct search algorithms is lack of a coherent mathematical analysis, particularly for solving constrained optimisation problems [144], [148].

3.3.2 Evolutionary Methods

Evolutionary optimisation algorithms are stochastic population based search methodologies which are usually inspired by some natural or physical principles [140]. The searching process of evolutionary algorithms begins from generating a random population of solutions within the design space. Thereafter, these populations are updated iteratively using different operators, in order to create new populations (i.e. each evolutionary algorithm has its own operators). This iterative process continues until a termination criterion is fulfilled [149]. The main reasons for popularity of evolutionary algorithms over classical optimisation methodologies can be summarised as following [140], [149], [150]:

- Evolutionary algorithms exhibit a global search process: given that the search process of evolutionary algorithms begins by generating diversely distributed solutions all over the design space, there is more chance that evolutionary algorithms discover all the regions of interest.

- Evolutionary algorithms do not require gradient information: the search process of evolutionary algorithms is based on direct search procedures (i.e. instead of using gradient information). Consequently, this algorithm can be applied for a wider range of complex engineering problems where the searching process might require overcoming a plethora of difficulties (e.g. non-differentiable objectives and constraints, infeasible regions, multiple optima, etc.).

- Evolutionary algorithms deliver more than a solution at each iteration: evolutionary algorithms are searching using a population of solutions at each iteration (i.e. unlike most of the classical methods which are delivering a solution at each iteration), which has a number of advantages as:

  - It enables using parallel computation to reduce the overall computational cost.

  - It facilitates the algorithm to determine multiple optimal solutions.

- Evolutionary algorithms exhibit a flexible searching process: evolutionary algorithms are employing a number of stochastic operators to update the initial solutions (i.e. unlike the classical methods which use deterministic operators). These operators can be tuned up for each optimisation problem to achieve the desired effect of operators.

Although evolutionary algorithms provide the opportunity to select appropriate operators for a specific optimisation problem, this flexibility can also be a potential shortcoming, given that each optimisation problem demands its own tuning of operators [151]. Moreover, despite the fact that using a population of solutions can increase the possibility of finding the global optimum, it results in a higher computational cost [140], [149].

In the following sections, two of the most popular evolutionary algorithms for solving SOO problems, Genetic Algorithm (GA) [137], [152], [153] and Particle Swarm Optimisation (PSO) algorithm [154], [155], are reviewed in details.

3.3.2.1 Genetic Algorithm (GA)

Genetic algorithm (GA), which is associated with Darwin's theory 'survival of fittest', was originally proposed as an optimisation algorithm by John Holland in 1960s [150]. This algorithm is a population-based stochastic search method inspired from genes behaviour, which is one of the most robust random search methods due to the element of directed-search [156]. Accordingly, GA has been broadly used as an alternative to the classical optimisation algorithms for solving complex engineering optimisation problems [157].

There are different explanations for describing genetic algorithm based on natural science. A simple explanation is that through the process of evolution all species are trying to adapt to the rapidly changing environment around them. Therefore, in this environment the species with a faster adoption process have more chance to survive. This process happens inside the DNAs (Deoxyribo-Nucleic Acid) which carry the genetic information in a cell. The genes inside each DNA contain a specific trait of the organism. During reproduction, each set of genes from a parallel cell combines with the second parallel set to generate a new gene set which contains a new organism trait. This process of reproduction is called Crossover. Sometimes there is a possibility of errors in copying genes from the parallel sets which is known as Mutation. Although mutation seems to be unbeneficial for the reproduction process, it can sometimes be substantially influential in species evolutionary changes [156].

GA optimisation algorithm is developed based on the same principals [158], as represented in Figure 3.1. The GA algorithm starts with generating initial random populations which are represented as coded strings of fixed length. These populations (i.e. also known as individuals) are representing the potential solutions. At each iteration of the algorithm, the fitness of each generated population is evaluated based on the objective function. Afterwards, GA operators (i.e. 'selection', 'crossover' and 'mutation') are employed to combine two of the randomly selected individuals (i.e. called parents) together and generate a new set of individuals (i.e. called children). This evolution process continues until a termination criterion is fulfilled. Consequently, only the individuals with better fitness value can survive.

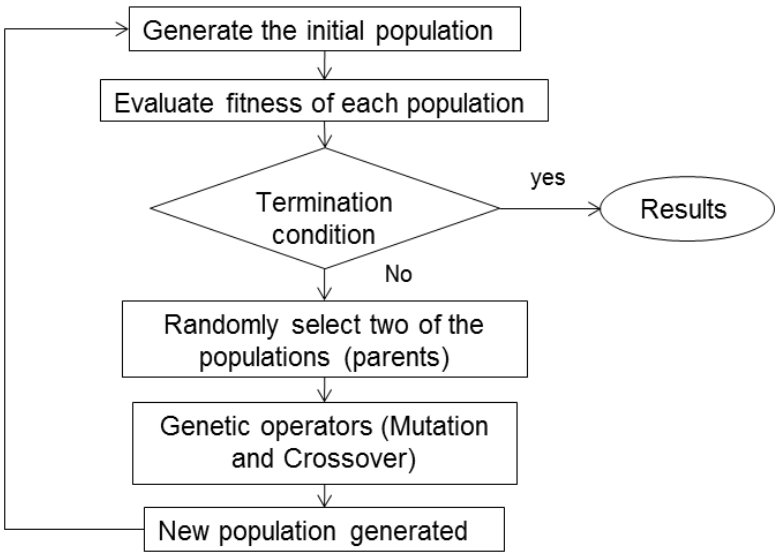


Figure 3.1: Simple GA procedure

In this section, the main components of GA algorithm are explained as following:

I. Genetic representation.
II. Genetic operators: selection, crossover and mutation.
III. Genetic parameters: elitist, population size, crossover rate and mutation rate.

I. Genetic representation

In this algorithm, the possible optimal solution for optimisation parameters is represented by a population (also called 'individual') using an encoding technique. The most utilised encoding schemes for a standard GA algorithm are 'Binary' (i.e. as strings of 0s and 1s) [158], [159], 'Gray' (i.e. as strings of 0s and 1s where two successive values differ in only one binary digit, such as 00 and 01) [160], and 'Real' (i.e. as strings of real-valued numbers) [160].

II. Genetic operators

The techniques used to combine the parent individuals and evolve them into new individuals (children) are called genetic operators. The main operators of a classic GA algorithm are: selection, crossover and mutation [159]:

a) Selection

'Selection' operator defines the method of selecting the parent individuals in a set of populations. Different selection methods have been introduced in literature, such as 'Tournament' [152] and 'Biased Roulette Wheel' [159]. In 'Tournament' selection strategy, a number of individuals are randomly selected based on the 'Tournament Size', and the individual with the best fitness is selected as a parent. However, in 'Biased Roulette Wheel' technique, the parent individuals are selected in a way that the individuals with better fitness values have proportionally more chance to be selected.

b) Crossover

'Crossover' operator is the technique used for combining the parent individuals, in order to create the child individuals [161]. There are different methods of crossover for the classical GA algorithm such as, uniform crossover, arithmetic crossover, single-point crossover, two-point crossover, partially-mapped crossover and gene-rank crossover [158], [159].

c) Mutation

This operator swaps two randomly selected elements of an individual [11], [158].

III. GA parameters

'GA parameters' are the effective parameters on the GA algorithm performance, which would require tuning for each specific optimisation problem to deliver an optimum solution. Grefenstette [162] discussed the effects of these parameters on GA convergence behaviour. In this section, some of these parameters are explained:

a) Elitist Strategy:

This GA parameter determines the number of individuals with the best fitness values, i.e. called 'Elite individual', to be transferred to the next iteration without any change. Thus, individuals with the best fitness value are not lost during subsequent generations, which consequently would assure a smooth convergence [163].

b) Population Size:

This parameter determines the number of individuals at each iteration, which can have a large influence on the solution quality. As an example, if the population size is small, there is a higher possibility that the algorithm would not discover all the design space and converges to a local optimum solution. On the other hand, a larger population size equals to more operating procedures, which would consequently results in a slower convergence rate [162].

c) Crossover Rate:

This parameter determines the number of individuals participating in the crossover operation. The potential pitfall of a small crossover rate is the higher possibility of an ineffective searching process, and thus entrapment in a local optimum. Similarly, a high crossover rate can have a destructive effect on GA performance because of dropping high performance individuals at each iteration [156], [160].

d) Mutation Rate:

This parameter determines the number of individuals perturbed in its own vicinity using the mutation operator [140]. A small mutation rate is usually preferred in order to introduce some extra variability into the population. Generally, a large mutation rate can interrupt the algorithm performance by adding a large number of random variables [158], [160].

Grefenstette [162] reported that GA algorithm performs well with a small population size, a high crossover rate and a low mutation rate. However, by increasing the population size, a lower crossover rate is preferred. It is noteworthy that the interaction among GA parameters is very complex and is usually dependent on the fitness function [151].

3.3.2.2 Particle Swarm Optimisation (PSO) Algorithm

PSO algorithm was originally developed by Kennedy and Eberhart [154], [155] with the inspiration of simulating the movement of organisms in a bird flock or fish school. PSO is a population based stochastic search algorithm, principled on the idea of representing each of the possible optimum solutions by a particle in a swarm (i.e. also called agent) which has its own position and velocity vector

[164]. Each agent has a memory containing information about the location of its personal best fitness value (P_{best}) and the location of the agent with the best fitness value in swarm (g_{best}) [164]. Accordingly, each agent can use the existing knowledge about the optimisation landscape to accelerate the convergence to the global solution. For a *n*-dimensional optimisation problem, velocity in dimension *n* can be modified using the following equation [165]:

Velocity update: $$v_n^{k+1} = wv_n^k + c_1 r_1\left(g_{best,n} - x_n^k\right) + c_2 r_2\left(P_{best,n} - x_n^k\right)$$ Equation 3.5

where *v* is the velocity, *k* indicates the iteration, c_1, c_2 are constants defining the relative pull of g_{best} and P_{best} respectively, *w* denotes the inertial weight, and r_1, r_2 are random numbers between 0 and 1. It is noteworthy that c_1, c_2 and *w* parameters might require tuning to ensure a robust convergence behaviour by the algorithm.

The agent's position in dimension *n* is also updated as follows:

Position update: $$x_n^{k+1} = x_n^k + v_n^{k+1} \times \Delta t$$
$$LB \leq x_n \leq UB$$ Equation 3.6

where *x* is the position, *LB* and *UB* are the lower and upper boundary limits for *x* in dimension *n*, and Δt denotes the time interval which is usually set to 1 [164].

The process of evaluating agents and updating positions and velocities continues until a termination condition is fulfilled. Figure 3.2 illustrates the PSO algorithm workflow.

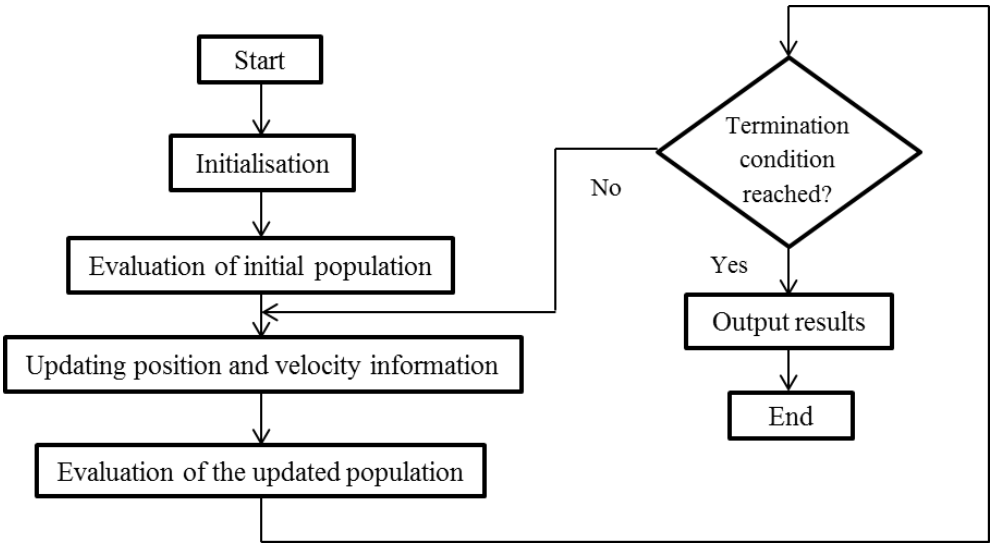


Figure 3.2: Flowchart of PSO algorithm

Several benchmarks are available which have studied PSO behaviour for solving complex engineering problems. Some of the main advantages of PSO algorithm over similar evolutionary techniques, such as *GA*, are [166]:

- PSO has a simpler structure and rather fewer parameters to be adjusted.
- PSO can efficiently maintain the diversity of swarm.
- PSO uses a more effective memory technique, which would accelerate the algorithm convergence.

3.3.3 Summary of SOO Methods

Table 3.1 summarises the main characteristics of the reviewed SOO algorithms, which can help to select the most appropriate algorithm.

Method	*Global or Local optimisation algorithm?*	*Sensitive to the initial search point?*	*Requires computing the gradient?*	*Requires tuning of parameters?*	*Convergence speed?*
Gradient Based	Local	Yes	Yes	No	Fast
Direct Search	Local	Yes	No	No	Fast
GA	Global	No	No	Yes	Slow
PSO	Global	No	No	Yes	Medium

Table 3.1: Characteristics of SOO algorithms

3.4 Multi-Objective Optimisation (MOO) Algorithms

The main target of multi-objective optimisation algorithms is to explore the efficient trade-off solutions in design space, also known as Pareto optimal solutions [147], for which improving one objective results in degrading the other one. Pareto solutions are also called non-dominated solutions in a sense that they are not dominated by other points in the objective space [140], [153], [167], [168]. Dominance term is given to the solutions which are not only strictly better than other solutions in at least one objective, but also not worse in all objectives [153]. Figure 3.3 shows an example of Pareto frontier for two objectives. In this example, the minimum value for objective 1 (f_1^*) is achieved at top left while the minimum value for objective 2 (f_2^*) is obtained at bottom right. In this case, for selecting a Pareto solution on the frontier to optimise both objectives, a solution with a smaller f_1 value has a larger f_2 value, and vice versa. The intersection of best values for f_1 and f_2 is called Utopia point which is impossible to achieve [34].

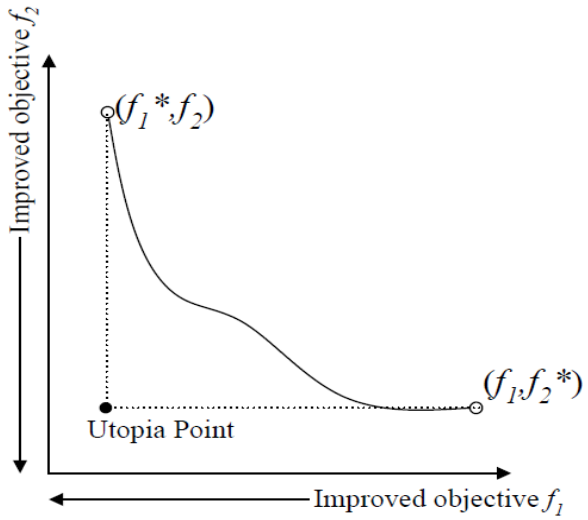


Figure 3.3: Objective function space of a two-objective function optimisation [34]

Similar to the single objective optimisation algorithms, multi-objective optimisation algorithms can be reviewed in two main categories [167], [168]: 'classical methods' and 'evolutionary methods'.

3.4.1 Classical Methods

Classical methods are mostly principled on the idea of scaling multiple objectives into a single objective optimisation problem and using a classical

SOO method to solve it. There are two possible strategies for generating a Pareto frontier using the classical methods [149]: 'one-at-a-time' and 'simultaneous'.

3.4.1.1 'One-at-a-time' Strategy

In this strategy, a parametric scaling technique is used to convert the multiple objectives to one 'master' objective function [149], and execute the single objective optimisation problem for several times with the aim of finding a Pareto solution at each time [141].

Although this method can most likely deliver a Pareto optimal solution, it is difficult to maintain diversity of Pareto solutions within the objective space [153], [169]. Moreover, executing several independent optimisation problems without a learning process about the design space (e.g. infeasible regions) is not an efficient searching process [149].

In this section, two of the widely used MOO algorithms which are formulated based on the 'One-at-a-time' strategy are reviewed [141].

I. 'Weighted sum' method: This method is based on scaling all the objectives, and combining them linearly by specifying a weight parameter (*w*) to each objective based on its importance [141]. This method can be mathematically defined as:

$$\begin{array}{lll} \textit{Minimise} & \sum_{m=1}^{M} w_m f_m(X) & \\ \text{with respect to} & X & \\ \text{subject to} & g_j(X) \leq 0 & j = 1,2,\dots,J \\ & h_k(X) = 0 & k = 1,2,\dots,K \\ & LB_i \leq x_i \leq UB_i & i = 1,2,\dots,n \end{array} \qquad \text{Equation 3.7}$$

Although implementation of weighted sum method is simple, there are some difficulties with this method [141]. The main shortcoming is that finding the best weight vector which represents the real objective function is challenging and a priori selection of weights cannot guarantee that the final solution is optimal. Moreover, this method cannot find a uniformly distributed Pareto-frontier even with a uniform spread of weight vectors [149].

II. ‘Bounded objective function’ method: In this method, the most important objective is defined as the objective function (f_s) while the other objectives are converted into boundary constraints in the objective space. These boundary limits (L_m and U_m) determine the acceptable tolerance for the converted objectives. This method can be mathematically formulated as [141]:

$$\begin{array}{lll} \textit{Minimise} & f_s(X) & \\ \text{with respect to} & X & \\ \text{subject to} & L_m \le f_m(X) \le U_m & m = (1,2,\dots,s-1,s+1,\dots,M) \\ & g_j(X) \le 0 & j = 1,2,\dots,J \\ & h_k(X) = 0 & k = 1,2,\dots,K \\ & LB_i \le x_i \le UB_i & i = 1,2,\dots,n \end{array} \qquad \text{Equation 3.8}$$

Implementation of ‘bounded objective function’ method is simpler than the ‘weighted sum’ method, since no scaling technique is required for the objective function [141]. However, the main drawback of ‘bounded objective function’ method is the infeasibility of optimisation problem when the boundary conditions on the objectives are not appropriate. Also, in this method the final supervision of a designer is essential before the final conclusion [141].

3.4.1.2 ‘Simultaneous’ Strategy

In ‘Simultaneous’ strategy, multiple Pareto optimal solutions can be explored in a single simulation run [149]. Various algorithms have been developed to perform simultaneous multi-objective optimisation. In this section, Normal Boundary Intersection (NBI) algorithm [170], which has been frequently used in automotive engineering [24], is reviewed.

NBI algorithm was originally developed by Das and Dennis [170] to define the Pareto frontier for nonlinear multi-objective optimisation problems. This algorithm is capable of delivering uniformly spread Pareto optimal solutions by employing a scaling technique [149]. This scaling scheme is fundamentally based on the idea that a uniform spread of parameters results in a nearly uniform Pareto solutions on the Pareto frontier [170]. To describe this technique

mathematically, consider $F^* = (f_1^*, f_2^*, \dots, f_M^*)^T$ to be the vector of optimum objective values for a multi-objective optimisation problem with *M* objectives and *n* variables, as given in Equation 3.1, which are obtained at individual optimum solutions x_i^*, $i = 1,2, \dots, n$. So, the simplex constructed by the solutions' convex hull, i.e. including the individual optimums, can be expressed by [149]:

$$\emptyset\boldsymbol{\beta} \qquad \text{where} \qquad \emptyset = (F(x_1^*), F(x_2^*), \dots, F(x_M^*))$$
$$\beta = \{(b_1, b_2, \dots, b_M)^T \mid \sum_{i=1}^{M} b_i = 1\} \qquad \text{Equation 3.9}$$

Consequently, a uniformly distributed Pareto solutions can be found by a proper setting of β vector [170]. The NBI sub-problem to define the Pareto solution for a set β can be shown as [149]:

$$\begin{aligned} &\textit{Maximise} && t \\ &\text{with respect to} && x, t \\ &\text{subject to} && \emptyset\beta + t\hat{n} = F(x) \\ & && h(x) = 0 \\ & && g(x) \leq 0 \\ & && LB \leq x \leq UB \end{aligned} \qquad \text{Equation 3.10}$$

where $\hat{n}$ denotes the pointing direction to the origin at $\emptyset\beta$, *h* is the equality constraint, *g* is the inequality constraint, and *LB* and *UB* define the lower and upper limits for the design variables, respectively. Therefore, this equation gives the maximum value for *t*, where the corresponding design variable is the Pareto solution x^*. It is noteworthy that this scaling technique is independent of the relative scales of the problem objectives [149].

NBI algorithm has been broadly used for various automotive applications [24], mostly due to availability of this algorithm in MATLAB Model Based Calibration (MBC) Toolbox [171]. However, this algorithm is not always a robust approach to find the Pareto frontier, given that this algorithm might converge to non-Pareto optimal solutions for non-convex problems [149]. Another limitation of this algorithm is that for the complex problems with more than two objectives (i.e. more than 2 dimensions), NBI cannot determine the extreme optimum values for some of the dimensions [172].

3.4.2 Evolutionary Methods

To overcome the drawbacks of classical methods in solving multi-objective optimisation problems, a number of evolutionary algorithms have been developed [168], [173]–[175]. Searching process of evolutionary algorithms is population based which involves a number of solutions (i.e. populations) at each iteration. Therefore, this can help the evolutionary algorithms to determine multiple non-dominated solutions in a single simulation run [149]. In this section, Elitist Non-dominated Sorting Genetic Algorithm (known as NSGA-II) [167], which is the most frequently used evolutionary algorithm for solving multi-objective optimisation problems, is explained in details.

NSGA-II [167] algorithm attempts to solve multi-objective optimisation problems by delivering a uniformly distributed Pareto frontier. NSGA-II procedure is illustrated in Figure 3.4 [140]. This iterative population-based algorithm starts by taking a sample of possible solutions P_t (i.e. called parent), followed by creation of a new solution Q_t (i.e. called child) by applying the usual genetic operators (e.g. crossover and mutation) on the parent population. Subsequently, these two populations are combined together to form a new population R_t. Size of the new population R_t is twice the size of the original populations. The next step is to evaluate the objective functions for the Pareto solutions stored in population R_t, in order to divide this population into different non-dominated classes. The new population is then created by inserting the non-dominated classes into the new population, one after the other. The insertion process uses an elitist principle to help the best Pareto points survive at each iteration. Given that the size of R_t is twice the size of the new population, not all the solutions can be accommodated. Therefore, the non-dominated Pareto solutions which cannot be accommodated will be deleted. However, it is usually the case that the last allowed non-dominated class has more solutions than the remaining slots. In this situation, instead of arbitrary deletion of the Pareto solutions in the last allowed non-dominated class, the solution with larger *crowding distance value* will be accommodated. Crowding distance value indicates the empty objective space around a solution which is not occupied by any other solution [140]. The described process continues in the next iterations until one of the termination criteria is met.

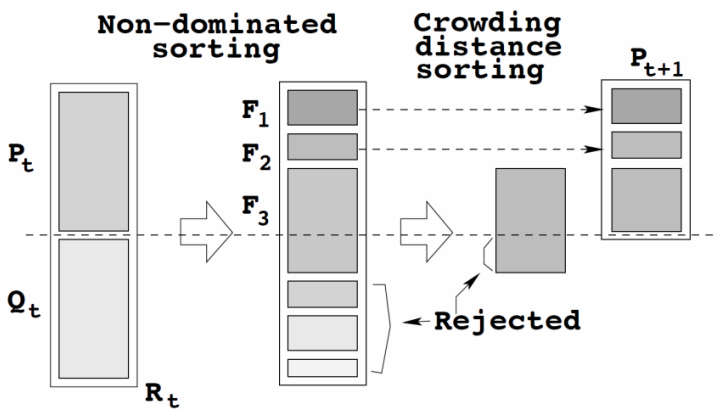


Figure 3.4: NSGA-II procedure [140]

The main advantages of NSGA-II compared to the other developed multi-objective evolutionary algorithms are [137], [153], [173], [176]:

- Convergence to an optimum solution is comparatively fast.
- Diversity of Pareto solutions on the objective space is controllable by crowding distance parameter.

However, one of the potential shortcomings of NSGA-II algorithm is that the performance of this algorithm can be affected by the crowding distance parameter [167].

Table 3.2 presents some of the main developments in evolutionary algorithms to solve multi-objective optimisation problems. This table summarises the techniques used by each algorithm to facilitate the convergence, and the corresponding advantages and disadvantages.

3.4.3 Summary of MOO Methods

Selecting the best multi-objective optimisation algorithm can be a difficult task and is thoroughly dependent upon the user's preferences [141]. The characteristics of the reviewed MOO algorithms in this section are summarized in Table 3.3.

Algorithm	Technique	Advantages	Disadvantages
Multi-Objective Genetic Algorithm (**MOGA**) [174]	- Non-dominated ranking scheme used to create the next potential Pareto solution. - Function of distributing Pareto solutions is based on the crowding distance within objective space.	- Delivering a non-dominated Pareto frontier. - Delivering uniformly distributed Pareto solutions.	Algorithm robustness is dependent on the sharing function.
Non-dominated Sorting Genetic Algorithm (**NSGA**) [168]	- Non-dominated sorting scheme used to create the next potential Pareto solution. - Function of distributing Pareto solutions is based on the crowding distance within design space.	- Delivering a non-dominated Pareto frontier. - Delivering uniformly distributed Pareto solutions.	Algorithm behaviour can be affected by changing the crowding distance.
NSGA-II [167]	- Elitist Non-dominated sorting scheme used to create the next potential Pareto solution. - Function of distributing Pareto solutions is based on the crowding distance within objective space.	- Fast convergence. - Delivering a non-dominated Pareto frontier. - Delivering uniformly distributed Pareto solutions.	Algorithm behaviour can be affected by changing the crowding distance.
Sub-Population Genetic Algorithm (**SPGA**) [175]	- Elitist scheme used externally to save the best Pareto solutions. - Uses clustering.	- Less iteration is required due to clustering. - Delivering a non-dominated Pareto frontier. - Delivering uniformly distributed Pareto solutions.	Algorithm is computationally expensive, due to external storing and data clustering.

Table 3.2: Summary of evolutionary algorithms

Method	*Always provides a Pareto optimal point?*	*Can provide all the Pareto optimal points?*	*Can provide a uniformly distributed Pareto frontier?*	*Involves weights?*	*Depends on function continuity?*	*Use utopia point?*
Weighted Sum	Yes	No	No	Yes	Yes	Yes
Bounded Objective	Yes	No	No	No	Yes	No
NBI	Yes	Yes (*)	Yes	No	Yes	No
NSGA-II	Yes	Yes	Yes	No	No	No

(*): NBI can find all the Pareto-optimal solutions when the Pareto front is convex.

Table 3.3: Characteristics of MOO algorithms

3.5 Multidisciplinary Design Optimisation (MDO)

The Multidisciplinary Design Optimisation (MDO) methods have been developed following the need for better optimisation frameworks and strategies to handle the high dimensional engineering problems, while addressing the complex couplings between system control variables. MDO frameworks, which are originally traced back to an extension of structural optimisation in order to include other disciplines [177], are a field of engineering with the main focus on using numerical optimisation for designing complex engineering systems which are governed by mutually interacting physical phenomena (subsystem or discipline). Interacting disciplines is commonplace for many engineering applications such as modern aircraft [34], [35], [178] and automotive vehicles [41], [179], [180].

MDO involves development of an engineering disciplinary decomposition in concert with the problem formulation to describe the interacting phenomena of a complex system. Solving such complex optimisation problems requires consideration of a methodology that organises the discipline analysis models, responses surface models (if any), and optimisation algorithms in a way that an optimal solution is achieved while the consistency of the system is maintained [34]. The behaviour of each subsystem is usually modelled using a discipline analysis, such as Polynomial models [80], [120] or RBF models [49], within the range of discipline parameters.

The main motivation of using MDO frameworks for complex optimisation problems is that these structural optimisation approaches consider not only the performance of each discipline but also their interactions. This can lead to a more accurate representation of system behaviour in order to deliver an overall optimum solution [27], [35].

However, the main challenges of employing MDO framework are:

- Organising the coupling among the system components can be a complex process subject to the optimisation problem, given that the discipline analysers are governed interdependently and one discipline's output might be required as an input in another discipline [27].
- Defining the system objectives and constraints can be challenging since these functions are based on both the design variables and coupling variables (i.e. outputs from the discipline analysers) [181].

Several MDO architectures have been introduced to address these challenges. In general, these frameworks can be divided into two main categories:

1. **Single-level MDO architectures** (also called *monolithic* [27]): which are solving the system problem through a single optimisation problem, such as Simultaneous Analysis and Design (SAND) [37], Individual-Disciplinary Feasible (IDF) [39], and Multi-Disciplinary Feasible (MDF) [182] methods.
2. **Multi-level MDO architectures** (also called *distributed* [27]): which are solving the system problem through partitioning the problem into a number of sub-problems (i.e. each consists of a subsection of variables and constraints), such as Concurrent Subspace Optimisation (CSSO) [183], Bi-Level Integrated System Synthesis (BLISS) [184], Collaborative Optimisation (CO) [35], and Analytical Target Cascading (ATC) methods [40], [41].

In this section, some of the most well-known single-level and multi-level MDO architectures will be explained in detail, followed by a comparison of their performances. However, before introducing the mathematical formulation of each approach, the notations used to present these optimisation methods will be presented.

3.5.1 Terminology and Mathematical Notation

In this chapter, similar notations are used to introduce the MDO approaches, in order to compare the methodology each architecture uses to handle the same MDO problem. These notations are summarised in Table 3.4.

In this notation list, design variable ***x*** is the quantity which is under the explicit control of the optimiser. If the design variable pertains to a single disciple, design variable is local $\boldsymbol{x_i}$ and if the variable is shared with more than a discipline, it is called shared variable $\boldsymbol{x_0}$. The combination of both local and shared variables is defining the design variable vector.

Coupling variables ***y***, also known as *linking* variables, are the variables exchanged between the disciplines to model the interactions of the system. In many of the MDO approaches, copies of coupling variables $\widehat{\boldsymbol{y}}$ are transferred to different disciplines in order to analyse the disciplines in parallel. These copies are also known as *target* variables [27].

Discipline analysis ***R*** is the simulation used to model behaviour of a subset (or discipline) of the multidisciplinary system. Each discipline analysis can be consisted of a system of equations itself, where the output of these equations defines the discipline responses (also called as state variables $\bar{\boldsymbol{y}}$). $|\boldsymbol{R}|$ denotes the mapping between y and $\bar{y}$ in residual format.

For each optimisation problem formulation, ***F*** denotes the objective function, which can itself be consisted of a number of equation responses, subject to a number of equality ***h*** and inequality ***g*** constraints. The combination of all equality and inequality constraints are shown by ***c***, *c* = [*g h*]. Consistency constraints $\boldsymbol{c^c}$ are also formulated in the optimisation problem function to guarantee the consistency between the linking variables input and output: $c_i^c = \widehat{y_i} - y_i$.

Symbol	Definition
x	Vector of design variables
y	Vector of coupling variables
$\bar{y}$	Vector of state variables
g	Vector of inequality constraints
h	Vector of equality constraints
c	Vector of all constraints
c^c	Vector of consistency constraints
F	Objective function
f	Vector of objective values
R	Governing equations of a discipline analysis
$\|R\|$	Governing equations of a discipline analysis in residual format
N	Number of disciplines
$()_0$	Functions or variables shared by more than a discipline
$()_i$	Functions or variables at discipline *i*
$()_s$	Functions or variables at system level
$()_{ss}$	Functions or variables at subsystem level
$()^*$	Functions or variables at their optimum
$\widehat{()}$	Independent copies of variables transferred to other disciplines
$\widetilde{()}$	Approximation of a given function

Table 3.4: Mathematical notations of MDO formulations

3.5.2 Single-Level Strategies

The most fundamental single-level optimisation formulation, from which all the other single-level MDO approaches are derived, can be shown as following [27]:

$$
\begin{aligned}
&\textit{Minimise} && f_0(x,y) + \textstyle\sum_{i=1}^{N} f_i(x_0, x_i, y_i) \\
&\text{with respect to} && x, \hat{y}, y, \bar{y} \\
&\text{subject to} && c_0(x,y) \le 0 \\
& && c_i(x_0, x_i, y_i) \le 0 && \text{for } i = 1, \dots, N \\
& && c_i^c = \hat{y}_i - y_i = 0 && \text{for } i = 1, \dots, N \\
& && \left| R_i\left(x_0, x_i, y_i, \bar{y}_i, \hat{y}_{j \neq i}\right) \right| = 0 && \text{for } i = 1, \dots, N
\end{aligned}
\qquad \text{Equation 3.11}
$$

This general formulation of single-level design optimisation is known as *All-At-Once (AAO)* problem [39] in a sense that all the problem variables (i.e. design variables, state variables, and input and output coupling variables) are optimised at once using a single optimiser.

Given that the general AAO formulation can be simplified without compromising the algorithm's performance, the AAO optimisation formulation is never solved in this format. Accordingly, three single-level MDO architectures have been introduced in literature [34], [37], [39] which are derived from Equation 3.11 depending on which equality constraint is removed:

1- **Simultaneous Analysis and Design (SAND)** [37]**:** developed through eliminating the consistency constraint $\boldsymbol{c_i^c}$ from the AAO formulation.

2- **Individual Disciplinary Feasible (IDF)** [27]**:** formulated by removing the discipline analysis constraint $|\boldsymbol{R_i}|$ from the AAO formulation.

3- **Multidisciplinary Feasible (MDF)** [182]**:** derived from the AAO formulation by reducing both consistency and discipline analysis constraints.

In the next subsections, the formulation of these architectures will be presented along with explanation of advantages and disadvantages of each approach. It is followed by relative comparison of architectures' performances.

3.5.2.1 Simultaneous Analysis and Design (SAND)

The SAND architecture is yielded on the idea of simplifying the AAO formulation through employing a single group of coupling (or linking) variables. Therefore, no additional consistency constraint is required for this formulation. The SAND formulation can be illustrated as [37]:

Minimise	$f_0(x, y)$		
with respect to	$x, y, \bar{y}$		
subject to	$c_0(x, y) \leq 0$		
	$c_i(x_0, x_i, y_i) \leq 0$	for $i = 1, \dots, N$	
	$\lvert R_i(x_0, x_i, y, \bar{y}_i) \rvert = 0$	for $i = 1, \dots, N$	Equation 3.12

Some of the main features of the SAND formulation can be summarised as [37], [39]:

- This architecture has an absolute non-hierarchical nature.

- This algorithm is not necessarily restricted to optimise the multidisciplinary optimisation problems, i.e. it can also be implemented for single-discipline optimisation problems.
- Given that the discipline analysis is not required to be solved explicitly at each iteration, the SAND framework is relatively robust.

However, there are two major issues that affect the efficiency of this algorithm to be implemented in practice [27]:

- Considering that all the state variables and discipline analysis are still required at each iteration, the size of optimisation problem can lead to premature termination of the optimisation algorithm and delivering a potential infeasible solution.
- An excessive computational effort might be required to calculate the discipline analysis constraints in residual format.

3.5.2.2 Individual Disciplinary Feasible (IDF)

The IDF optimisation framework is based on a single system optimiser and a number of discipline analysers, which is formulated in a way that the system optimiser controls the interactions between disciplines [34].The IDF architecture is mathematically derived through elimination of the discipline analysis constraints (i.e. which also results in elimination of state variables) from the AAO formulation [39]. This simplification can be achieved by applying implicit function theorem to the discipline analysis constraints in order to represent y and $\bar{y}$ variables as function of x and $\hat{y}$. The IDF mathematical formulation can be represented as [181]:

$$
\begin{aligned}
&\textit{Minimise} && f_0(x,y) && && \\
&\text{with respect to} && x,\hat{y} && && \\
&\text{subject to} && c_0(x,y) \le 0 && && \\
& && c_i(x_0,x_i,y_i) \le 0 && \text{for} \quad i=1,\dots,N && \\
& && c_i^c = \hat{y}_i - y_i = 0 && \text{for} \quad i=1,\dots,N && \text{Equation 3.13}
\end{aligned}
$$

In this equation, the optimiser is working in respect to both design variables and coupling variable copies, whereas the consistency constraint preserves the system consistency [30], [39], [185], [186].

The main advantages of implementing IDF framework are [30], [39], [185]:

- Hierarchical nature of IDF formulation.
- Parallel nature of IDF formulation which offers improved compression of design process. The parallelisation is enabled through linking the disciplines using coupling variable copies $\hat{y}$ and consistency constraints c^c.
- Improved robustness since the consistency constraint is not required to be solved explicitly at each iteration.
- Relatively a faster convergence since the problem formulation is smaller than the SAND problem.

However, there are still some shortcomings that can influence the performance of an IDF formulation. Some of these limitations are [27]:

- Size of the optimisation problem can substantially affect the efficiency of the optimisation framework due to the large number of linking variables to be optimised. Although this limitation can be mitigated to some extent through grouping some of the linking variables, i.e. which principally reduces the information transfer among the disciplines.
- Implementations of gradient-based optimisation algorithms can potentially lead to an excessive computational process due to costly process of evaluating the objective and constraint function gradients.

3.5.2.3 Multidisciplinary Feasible (MDF)

MDF optimisation framework is obtained from removing both consistency constraints and discipline analysis constraints from Equation 3.11 [39].

$$\begin{aligned} &\textit{Minimise} && f_0(x, y) \\ &\text{with respect to} && x \\ &\text{subject to} && c_0(x, y) \leq 0 \\ & && c_i(x_0, x_i, y_i) \leq 0 \qquad \text{for} \quad i = 1, \dots, N \end{aligned} \qquad \text{Equation 3.14}$$

The MDF method has been successfully employed for different applications in industry, such as aerospace [182], [186]. Some of the advantages of using MDF architecture in comparison to the other single-level MDO approaches are:

- The optimisation problem is formulated in the simplest possible way, where the system optimiser is only responsible for controlling the design variables, objective function and design constraints [39].
- Given that the system optimiser solves the optimisation problem explicitly at each iteration, the MDF approach delivers a solution that always meets the consistency constraints even if the algorithm terminates prematurely. Consequently, this approach can be an efficient method if the target is to define a feasible solution (not particularly an optimum solution) within a limited time [27].

However, this framework has some limitations that may result in necessity of selecting other MDO approaches for complex engineering problems. Some of these drawbacks are [34], [182], [186]:

- MDF approach has an absolute non-hierarchical nature.
- The computational process can be very expensive due to requirements of more iterations for converging in strongly coupled systems. Therefore, considering the fact that there is no restriction for data communication between subsystems in MDF method, weakly coupled system can be more beneficial due to fast convergence.
- MDF approach cannot be parallelized, thus required time for design cycle can be noticeable.
- MDF is a low robust approach since the optimiser can easily fail if the analysis does not converge in an iteration. Consequently, gradient-based optimisation algorithms cannot ensure whether an optimum solution is achieved.
- MDF is not capable of finding the best solution while more than one analysis solution exists.

3.5.2.4 Comparison of Single-Level MDO Approaches

Table 3.5 illustrates the comparison of explained single-level MDO approaches. This table can help in understanding the advantages and disadvantages of each formulation, to select the most appropriate optimisation framework.

It is noteworthy that although Single-Level methods have shown an acceptable performance for many of the optimisation problems in practice, there are still some drawbacks that may have a profound effect on efficiency of these MDO architectures. Some of these weaknesses are [34]:

- Existing single-level formulations may not be successful in centralizing the whole system design. In a centralized system, the decision making power is concentrated in the top level and tight control is applied over subsystems.
- An efficient centralization of whole system may require excessive data communication for the single-level approaches.

Method	SAND	IDF	MDF
Satisfaction of governing equations	At each iteration	At each iteration	At each iteration
System consistency	Only at convergence	Only at convergence	At each iteration
Expected speed	Medium	Medium	Slow
Optimiser variables	x, y	$x, \hat{y}$	x
Robustness	High	High	Medium
Problem structure	Non- hierarchic	Hierarchic & non- hierarchic	Non- hierarchic

Table 3.5: Comparison of Single-Level MDO approaches

3.5.3 Multi-Level Strategies

Many MDO architectures have been introduced in literature which are formulated based on the idea of decomposing the large optimisation problems into smaller sub-problems, and reassembling the solutions to deliver an overall optimum solution. The primary motivation of implementing a Multi-level optimisation approach stems from the nature of problem handling in engineering-design environment, where a large system problem (e.g. designing an aeroplane) is generally solved through breaking down the system problem into a number of sub-problems (e.g. aeroplane wing, tail and engine), such that different groups work on separate sub-problems in isolation and control their own design variables based on their in-house expertise. These groups get updated from the other groups' progress periodically in order to modify their own design to satisfy the overall aspects of the design [181], [185].

In Multi-Level MDO approaches, subsystem level optimisers are introduced which principally change the system level optimiser duty. Accordingly, the main objective of system optimiser is to guide the whole procedure to the optimum system solution while coordinating the subsystem level interactions [185].

In general, Multi-Level MDO architectures can be reviewed within two primary categories [27]:

1. Multi-Level architectures which are principally based on decomposition of the MDF optimisation method, such as Concurrent Subspace Optimisation (CSSO) strategy [184] and Bi-Level System Synthesis (BLISS) method [38], [187].
2. Multi-Level architectures which are principally based on decomposition of the IDF optimisation approach, such as Collaborative Optimisation (CO) framework [33], [35] and Analytical Cascading Target (ATC) strategy [28], [40]–[42], [188], [189].

In the following subsections, some of the most important optimisation architectures from each of these two categories will be introduced.

3.5.3.1 Concurrent Subspace Optimisation (CSSO)

CSSO architecture, which is based on decomposing the MDF problem along the disciplines, was originally formulated to decompose large scale MDO problems into several independent sub-problems using disconnect set of variables [183]. Thus, this architecture includes all the design variables at each of the sub-problems.

In this architecture, a coordination problem is solved at the system level to define the “responsibility” coefficients to be transferred to each of the disciplines [27], [190]. These coefficients provide the required information regarding the preferred values for design variables to satisfy the nonlocal constraints. Accordingly, each discipline optimisation is handled independently within the system as whole, by using the transferred information. The whole system convergence is enhanced through calculation of a linear approximation of the global sensitivity at each iteration, and transferring it into the sub-problems.

Several version of CSSO architecture are available in literature [190]–[192]. In this section, the formulation proposed by Sellar *et al*. [191] is used to illustrate the CSSO architecture at system level and sub-problems, as shown in Equation 3.15 and Equation 3.16, respectively. In this formulation, the discipline analysers are replaced by the surrogate models. Therefore, for each discipline, the required information from the other disciplines to model the multidisciplinary interactions is provided by solving the surrogates of the other disciplines.

Minimise	$f_0(x, \tilde{y})$		
with respect to	x		
subject to	$c_0(x, \tilde{y}) \leq 0$		
	$c_i(x_0, x_i, \tilde{y}_i) \leq 0$	for $i = 1, \dots, N$	Equation 3.15

Minimise	$f_0(x, y_i)$		
with respect to	x_0, x_i		
subject to	$c_0(x, \tilde{y}) \leq 0$		
	$c_i(x_0, x_i, y_i) \leq 0$		
	$c_j(x_0, \tilde{y}_j) \leq 0$	for $j = 1, \dots, i-1, i+1, \dots, N$	Equation 3.16

The behaviour of CSSO architecture in dealing with complex engineering problems has been investigated in several articles [190]–[193]. The authors noted that CSSO framework has several fundamental shortcomings, such as the sensitivity of this architecture to the parameter selection, which can result in extensive tuning even for a noncomplex nonlinear problem. Another potential pitfall of this architecture is the need for all of the design variables at each subproblem, given that CSSO decomposes the problem into disjointed disciplines.

Several benchmarks are available which have investigated performance of CSSO framework in comparison with other MDO approaches, e.g. Tedford and Martins [194]. Generally, the results show that CSSO architecture is largely an ineffective approach to handle the complex engineering problems, given that this architecture requires many more function evaluations to converge to an optimal solution [27].

3.5.3.2 Bi-level Integrated System Synthesis (BLISS)

BLISS architecture [187] is also based on decomposition of the MDF problem along the disciplines, similar to the CSSO architecture. However, in this architecture instead of transferring all the design variables to the disciplines, the local design variables are assigned to sub-problems and the shared design variables are assigned to the system problem [38].

Essentially, BLISS architecture is based on using a gradient-based path within the design space (i.e. also called sensitivity analysis) by constructing a linear approximation of the original problem [184]. In this architecture, the variables bound are pre-defined by the user in order to avoid moving to the parts of design space where the approximation is not sufficiently accurate.

The BLISS problem formulation for the system problem and the sub-problems are given in Equation 3.17 and Equation 3.18, respectively.

Minimise	$(f_0^*)_0 + \left(\frac{df_0^*}{dx_0}\right)\Delta x_0$		
with respect to	Δx_0		
subject to	$(c_0^*)_0 + \left(\frac{dc_0^*}{dx_0}\right)\Delta x_0 \leq 0$		
	$(c_i^*)_0 + \left(\frac{dc_i^*}{dx_0}\right)\Delta x_0 \leq 0$	for $i = 1, \dots, N$	
	$\Delta x_{0_L} \leq \Delta x_0 \leq \Delta x_{0_U}$		Equation 3.17

Minimise	$(f_0)_0 + \left(\frac{df_0}{dx_i}\right)\Delta x_i$		
with respect to	Δx_i		
subject to	$(c_0)_0 + \left(\frac{dc_0}{dx_i}\right)\Delta x_i \leq 0$		
	$(c_i)_0 + \left(\frac{dc_i}{dx_i}\right)\Delta x_i \leq 0$		
	$\Delta x_{i_L} \leq \Delta x_i \leq \Delta x_{i_U}$		Equation 3.18

Given that the BLISS architecture relies on linear approximations, repeating the evaluation of objective and constraint functions are not required when the gradient is available [187]. However, reliance on the linear approximation can be a potential pitfall itself when the problem is highly nonlinear. For such a case, a proper definition of the variables bound can help the convergence, whereas it requires prior knowledge about the design space [27].

Many of the benchmarks available for original BLISS architecture [195], [196] argue that BLISS performance is not competitive comparing to the other MDO frameworks. The main drawback of this approach is the computational cost, which is the result of both the large number of discipline analysis and the high cost of calculating derivatives.

3.5.3.3 Collaborative Optimisation (CO)

The CO optimisation framework is a strategy to decompose the IDF problem along the discipline lines [27]. CO architecture is principled on independent optimisations at each discipline, whereas the disciplines are connected to each other through receiving copies of linking variables and sharing design variables (if there is any) at every iteration from the system level optimiser (i.e. copies of variables are also known as *target*s in many of references) [197]. Considering the capability of CO architecture to optimise the disciplines in parallel, combine with the CO's simple process of data-sharing among the disciplines, makes this approach a strong candidate for optimisation of the multi-attribute optimisation problems with a small number of sharing design variables [32], [33], [35], [186].

Accordingly, in CO framework, which is a bi-level hierarchal formulation, the duty of system-level optimiser is to minimise the design objective function, while the sub-system optimisers (i.e. at each discipline) are responsible to minimise the discrepancy between the system level targets and each discipline solutions in order to minimise the whole system inconsistency [32], [33], [35]. The optimisation formulation for the system problem is shown in Equation 3.19.

Minimise	$f_0(x, \hat{y})$	
with respect to	$x, \hat{y}$	
subject to	$c_0(x, \hat{y}) \leq 0$	
	$c_i^c = \|\hat{x}_i - x_i\|_2^2 + \|\hat{y}_i - y_i\|_2^2 = 0$	for $i = 1, \dots, N$ Equation 3.19

$\hat{y}$ is the copy of linking variables, and $\hat{x}$ is the copy of global design variables (i.e. copies of shared design variables between two or more disciplines $\hat{x}_0$) which are transferred to each discipline *i* as the target value to be met. These copies of variables are independent from each other which facilitate the CO architecture to solve the sub-problems in parallel. In this formulation, the

equality constraint c_i^c is responsible of preserving the system consistency trough ensuring both system level and subsystem level optimum solutions are agreeing on a single value.

The optimisation problem at discipline *i* is:

Minimise	$\|\hat{x}_i - x_i\|_2^2 + \|\hat{y}_i - y_i\|_2^2$	
with respect to	x_i, y_i	
subject to	$c_i(x_i, y_i) \leq 0$	Equation 3.20

Where at each discipline the objective is to meet the targets transferred from the system level, i.e. by minimising the discrepancy between the subsystem level solution and system level copies of variables, while the local constraints at each discipline are also satisfied. Subsequently, the subsystem level solutions for the linking variables and shared variables will be returned to the system level.

It is argued by Braun [197] that CO framework is mathematically equivalent to the IDF approach. In particular, if CO framework converges to a local solution where all the system level and subsystem level constraints are satisfied ($c_0 \leq 0$, $c_i \leq 0$, and $c_i^c = 0$) the solution must also be a local solution for the IDF approach.

Some of the benefits of CO approach in comparison with single-level strategies are [33]:

- The Optimisation process requires less system communication, which is due to capability of CO architecture to solve the discipline sub-problems independently.
- Performs well for both hierarchical and non- hierarchical MDO problems.

Regardless of these advantages, CO architecture has several major shortcomings that might affect the performance of this mathematical formulation for complex engineering problems. Some of these weaknesses can be summarised as:

(i) Convergence difficulty because of high number of equality constraints that can lead into infeasibility at sub-problems [34], [198]. The whole system

consistency is often endangered if the subsystem level returns a significantly different solution for the interdisciplinary variables [199].

(ii) Slow convergence of gradient-based optimisation search algorithms because the constraint gradients at an optimal solution are zero vectors, which denote a failure in the satisfaction of Karush-Kuhn-Tucker optimality conditions [200].

(iii) No convergent coordination strategy is defined to enable system problem decomposition to more than two levels of hierarchy [34].

In some cases, researchers have attempted to enhance the robustness of the CO architecture by addressing the above shortcomings [181], [198]. For example, some authors used gradient-free optimisers (e.g. genetic algorithm) instead of gradient based search algorithms to handle the troublesome constraints [201]. However, the gradient-free optimisers raised other issues such as increasing the number of required function evaluations.

Despite the mathematical limitations of the CO architecture, this optimisation approach has been frequently used in industry, mostly for the aerospace design applications [35], [186].

3.5.3.4 Analytical Target Cascading (ATC)

The ATC architecture was not originally designed as an optimisation framework to handle the multidisciplinary problems. This approach was developed as a strategy to propagate the desirable design targets (*T*) through a hierarchical system until a feasible design that can satisfy all the targets is achieved [42], [202]. Thus, if satisfying all the targets is unattainable, the architecture returns a design points which minimises the unattainability [27].

Typically, this partitioning based organisation of the problem matches the systems engineering design problem from a product development point of view [42]. Figure 3.5 provides an automotive illustration of the function based hierarchical decomposition, where the utility function associated with the customer requirement (e.g. ‘torque demand’) is mapped to a functional requirement for the vehicle systems (e.g. powertrain – for which the main functional aim is to generate torque), which in turn is iteratively cascaded to the

relevant subsystems (e.g. engine) and components (air and fuel intake and spark – for a gasoline engine). The system needs to be designed in a way that the customer demand is met at any time, which requires coordination of targets cascaded from the customer down through the system hierarchy to each component, as well as bottom-up re-balancing to ensure that subsystem requirements are met.

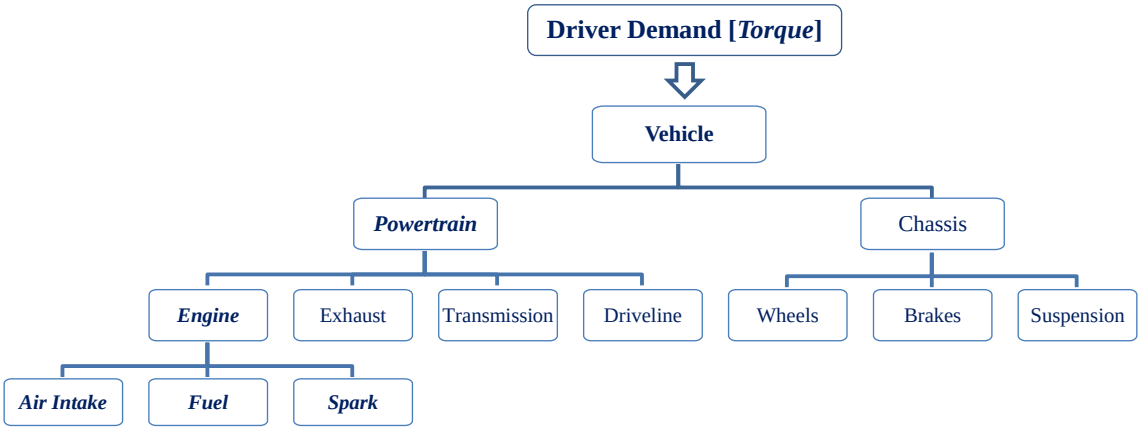


Figure 3.5: Illustration of Target Cascading in a vehicle systems engineering design

Figure 3.6 illustrates the functional hierarchy in the systems engineering cascade for the example in Figure 3.5, as well as a comparison between the single-level design optimisation problem formulation as 'all-at-once' (AAO) [34], discussed in Section 3.5.2, on the left, and ATC, on the right. The main benefits of target cascading are the reduction in the analysis cost and time by decreasing the dimensionality of the optimisation problem (compared to all-at-once optimisation), and at the same time maintaining the whole system consistency through the rebalancing-up [41], [42], [199].

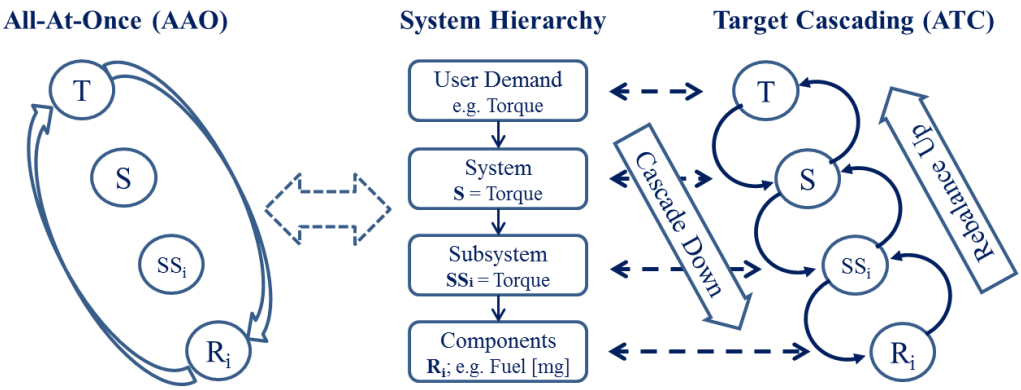


Figure 3.6: Illustration of AAO and ATC for a vehicle systems engineering design

The mathematical formulation for the standard ATC architecture at the system level can be illustrated as:

$$
\begin{aligned}
&\textit{Minimise} && \|T - f_0\|_2^2 + \sum_{i=1}^{N} \|\hat{x_i} - x_i\|_2^2 + \|\hat{y_i} - y_i\|_2^2 + \epsilon_1 + \epsilon_2 \\
&\text{with respect to} && x, \hat{y} \\
&\text{subject to} && c_0(x, \hat{y}) \leq 0 \\
& && \|T - f_0\|_2^2 \leq \epsilon_1 \\
& && \sum_{i=1}^{N} \|\hat{x_i} - x_i\|_2^2 + \|\hat{y_i} - y_i\|_2^2 \leq \epsilon_2
\end{aligned}
\qquad \text{Equation 3.21}
$$

T is the design target, f_0 is the system objective function, ‖ ‖ denotes a metric for discrepancy, $\hat{y}$ is the copy of linking variables, $\hat{x}$ is the copy of shared design variables, while ε_1 and ε_2 are deviation tolerances introduced to coordinate the subsystem level responses (as discrepancy between the target passed down and the response from the lower subsystems), and c_0 is the vector of inequality and equality constraints imposed at the system level.

The ATC cascade starts at the highest level of the system with a target, T, so the first optimisation is with respect to the discrepancy between the response and the target, i.e. $\|T - f_0\|$.

After solving the system level optimisation problem, the copies of shared design variables $\hat{x}$ and linking variables $\hat{y}$ are passed down to the subproblems at each discipline *i* (i.e. subsystems) as the design target, and the solution of the subsystem level optimisation, x_i and y_i, are returned to the system level [29], [41], [42], [199], [202]. In general, at each level in the hierarchy, the ATC optimisation problem formulation can be formulated as minimisation of the discrepancy between the cascaded design targets and the returned responses, subject to system design constraints being satisfied [29], [41], [42], [199], [202].

Given the hierarchical decomposition of the ATC architecture and the information communication strategy in the system through propagation of targets, ATC is effectively working with the same principles as other multi-level MDO frameworks. Principally, system level objective of a standard ATC method, i.e. minimisation of the discrepancy between system targets and system responses [42], [202], can be simply modified to solve standard MDO problems. Several modified approaches have been discussed in literature [189]

in which the ATC cascade is not necessarily started with a specified target, reflecting product development situations where a design target is not necessarily known a priori. In such situations, the system level response is minimised at the top-level of the hierarchy (or system level) while the system consistency is preserved using a penalty function (∅). Subsequently, the solution is cascaded down to the lower levels as the target. Accordingly, the mathematical formulation of ATC architecture at the system level can be shown by [203]:

Minimise	$f_0(x,\hat{y}) + \sum_{i=1}^{N} \emptyset_i(\hat{x}_i - x_i\ ,\ \hat{y}_i - y_i)$	
with respect to	$x, \hat{y}$	
subject to	$c_0(x,\hat{y}) \leq 0$	Equation 3.22

The subproblem formulation for discipline *i* is given by:

Minimise	$f_i(x_i,y_i) + \emptyset_i(\hat{x}_i - x_i\ ,\ \hat{y}_i - y_i)$	
with respect to	x_i, y_i	
subject to	$c_i(x_i,y_i) \leq 0$	Equation 3.23

$\emptyset_i$ denotes the penalty relaxation for discipline *i*, which is a function of discrepancy between the transferred target values for the variables ($\hat{x}_i$ and $\hat{y}_i$) and the optimisation variables (x_i and y_i). There are several types of penalty functions in literature, such as quadratic penalty function [204], Lagrangian relaxation [205], and augmented Lagrangian relaxation [206], which have been proposed to enhance the ATC convergence property by updating (i.e. increasing) the penalty weights until the desired consistency is achieved.

The performance of ATC architecture in comparison with other Multi-level MDO architectures (i.e. CSSO, CO and BLISS) is studied by de Wit and Van Keulen [207]. In this research, it is shown that the ATC performance is competitive (i.e. in both solution quality and computational efficiency) compared to the other tested multi-level approaches.

The ATC framework has been frequently applied in the automotive industry, for the field it was originally developed [27]–[29]. However, it also provides a useful

architecture for the optimisation problems in other fields, such as aerospace [30], [31].

3.5.3.5 Comparison of ATC and CO Multi-Level MDO Approaches

Both ATC and CO architectures have been frequently implemented in practice for optimisation of multi-attribute optimisation problems [27]–[29], [32], [33]. Fundamentally, it can be argued that both ATC and CO architectures are a strategy to decompose the IDF problem [27]. However, the main difference between CO and ATC formulations is that in ATC formulation the system consistency is ensured using a penalty function, while in CO the system consistency is maintained by an equality constraint. Another primary difference between ATC and other MDO approaches is the ability of ATC architecture to handle the multidisciplinary optimisation problems using a multi-level structure with more than two levels of hierarchy, i.e. by focusing on both targets and system variables.

There are some other differences between CO and ATC frameworks. One of the differences which sometimes cause confusion is the meaning of 'Target' for these approaches. For CO approach 'Target' refers to design and linking variables while in ATC 'Target' refers to shared design variables, linking variables and the higher level responses [34]. Table 3.6 summarises some of the main properties of CO and ATC strategies:

Property / Method	**ATC**	**CO**
Intended usage	Early product development (particularly in Automotive industry)	Multidisciplinary analysis & product design (particularly in Aerospace field)
Consistency constraints	Relaxed penalty function	Equality constraint
Optimiser variables	$x, \hat{y}, T$	$x, \hat{y}$
Problem structure	Hierarchic	General non-hierarchic
Number of top level optimisation	Multiple	Single
System consistency	Proven Convergence	------
Robustness	High	Medium

Table 3.6: Comparison of ATC and CO approaches

3.5.4 Hybrid Strategies:

Sometimes an optimisation problem cannot be fitted in a standard optimisation formulation. Mostly it is hard to describe the complex optimisation systems with so many subsystems in a traditional design optimisation framework. Therefore to implement a proper optimisation method, a framework is required which consists of serial and parallel structures at the same time. Accordingly, hybrid methods can be customized to fit the best MDO optimisation framework for a specific engineering problem. An example of a collaborative model of hybrid hierarchical systems is shown in Figure 3.7. This system consists of a hierarchic system, which contains two subsystems, and a non-hierarchic system that also consists of two subsystems [208]. Sometimes it is possible to change a hybrid architecture to a standard MDO formulation by introducing additional variables and constraints, but breaking the system communication may lead in disturbing the design team [34].

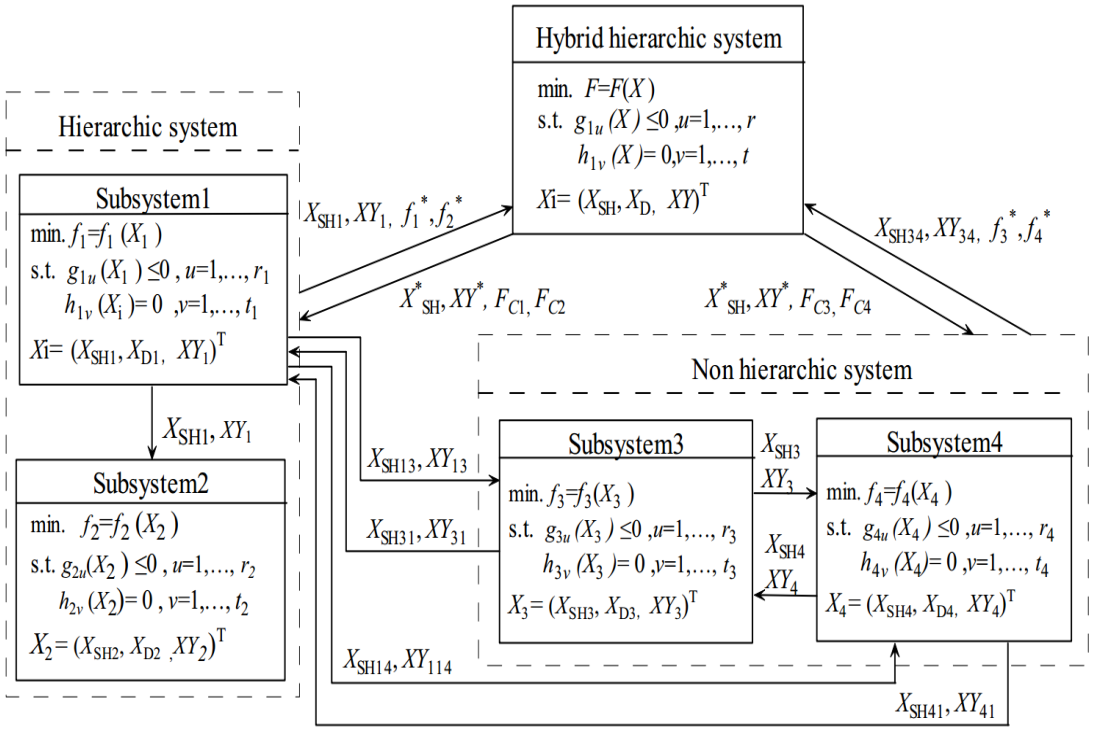


Figure 3.7: An example of a Hybrid MDO Framework [208]

3.6 Review of Optimisation Strategies for Engine Calibration Optimisation

In general, the complexity of the optimisation problem for the steady state calibration arises from the implicit two-stage structure of the Model Based Calibration (MBC) process, as illustrated by Figure 2.1:

(i) ‘Local’ optimisation stage, which aims to identify a set of ‘local’ (i.e. at each engine speed / load point) optimal solutions that satisfy local objectives and constraints. The local problem is often defined as a ‘trade-off’ optimisation between main objectives [7], such as NOx and particulate emissions for a Diesel engine [209].

(ii) ‘Global’ optimisation stage, which aims to identify a ‘global’ (i.e. over the engine speed / load operating envelope) solution from the local optimal sets. Generally, this is based on criteria associated with the overall targets for the engine or vehicle, such as overall fuel consumption and emissions over a specific drive cycle (e.g. NEDC emissions drive cycle) [210].

Historically, the focus of optimisation methods development in MBC is often placed on the local optimisation task. Both *single-objective* and *multi-objective* optimisation formulations have been frequently used in literature, aiming to implement an efficient algorithm to identify viable local solutions.

Millo *et al*. [211] formulated a turbocharged Gasoline Direct Injection (GDI) engine calibration problem using a single-objective optimisation approach, for both full load and part load. In this formulation, the calibration optimisation is formulated to maximise BMEP (Brake Mean Effective Pressure) in subject to Knock and brake torque constraints, aiming to define the optimum setting for calibration parameters (i.e. Air/Fuel ratio, spark angle and boost) at each minimap (engine speed / load operating condition). In this approach different types of gradient-based search algorithms were employed to solve the optimisation problem.

Similarly, Sheridan *et al*. [59] and Popovic *et al*. [212] proposed a single-objective formulation for calibration optimisation problem. Sheridan *et al*. [59] used a gradient based algorithm (Gradient descent algorithm) to maximize torque at each operating point in subject to exhaust temperature and engine stability constraints. In this investigation, the authors used two- stage modelling techniques with different types of approximation models, such as Radial Basis Function (RBF) models. In the same way, Popovic *et al*. [212] used variety of gradient based algorithms, such as Box and Wilson Steepest Descent (BWSD)

algorithm, to minimize Brake Specific Fuel Consumption (BSFC) at each minimap in subject to emission constraints. Popovic *et al*. [212] proposed this formulation for low and medium torque operating points to avoid the operating points with knock limitations.

Identifying a global optimal solution for the engine calibration problem is often reduced to an exploratory search of local optimal solutions sets. Considering the requirements to fulfil global objectives, such as fuel consumption and emissions over the drive cycle, and also to ensure that other engineering criteria are met by the calibration, such as the smoothness of the actuator maps linked to driveability attributes [209], emphasises the need to implement efficient multi-objective optimisation algorithms to identify viable local trade-off solutions, usually based on a Pareto optimal set [2], [21], [22].

Morton *et al*. [24] proposed to implement a multi-objective optimisation framework for a gasoline engine, using NBI search algorithm [170]. The main objective of this formulation was to minimize both BSFC and NOx emissions at each minimap subject to manifold pressure and exhaust temperature constraints. However, considering the dependency of NBI solution on the initial search point [170], this algorithm had to be run for several times to ensure that NBI was not trapped in a local optimum solution.

So, to improve the calibration optimisation solution, Vossoughi and Rezazadeh [23] proposed to employ evolutionary multi-objective algorithms. They formulated the gasoline engine calibration problem as the trade-off between BSFC and NOx, similar to Morton *et al*. [24], using two different evolutionary multi-objective algorithms (i.e. Distance-Based Pareto Genetic Algorithm and NSGA-II). Similarly, Kaji and Kita [21], Kaji *et al*. [213], and Singh *et al*. [2] employed NSGA-II evolutionary algorithm to optimise the steady state calibration of a gasoline engine.

Kaji and Kita [21] formulated the calibration optimisation problem as minimising fuel consumption and maximising torque at each minimap subjected to Combustion stability, to define the optimum solution for the each calibration parameter (i.e. . Fuel injection timing, Ignition timing, Air-fuel ratio and Valve control parameter). Also, Singh *et al*. [2] defined the optimum calibration setting

for the parameters (i.e. camshaft timings) through maximising torque and minimising SDNMEP (Standard Deviation of Net indicated Mean Effective Pressure), which is indicator of combustion stability, at each minimap with respect to the constraint on SDNMEP. In these papers, NSGA-II algorithm behaved robustly for determining the Pareto optimal solutions, as potential global solutions for the multi-objective optimisation problem, however; the convergence of this algorithm was slow.

Seabrook *et al*. [22] employed two Pareto optimal sets, (1) BSFC versus load and (2) BSFC versus combustion stability indicator, instead of using a single objective approach to minimise the BSFC subject to a combustion stability constraint. Using these Pareto frontiers, the global calibration solution was selected by manually comparing the local optimum candidates, with respect to the other additional calibration requirements (e.g. map smoothness requirement).

Although a multi-objective formulation, using Pareto optimal frontier, can empower the calibrator to manually select a feasible global optimum setting satisfying the calibrator preferences, it often results in difficulties with the global optimisation stage. This can be a very time consuming and iterative process, demanding calibration expertise in selecting a good set of global solutions, often requiring resampling from the local trade-off solutions set if global constraints cannot be met [43], and significant time and effort for the downstream calibration process. Attempts to address these difficulties for multi-objective approaches have been based on either:

(i) Development of ‘global’ response models [62], i.e. across the engine speed / load operating envelope, incorporating engine speed and engine load as variables. Given the increasing number of calibratable variables, this approach could require extensive testing effort to generate models of sufficient quality to support optimisation.

(ii) Combination of local and global optimisations in the same problem formulation. For example, Roudenko [214] suggested a multi-objective optimisation formulation to minimise fuel consumption (global optimisation) and noise (local optimisation) under the constraints for

global emissions (NO_x and Soot) for a Diesel engine. However, the main shortcomings of this approach are the increase in the search space dimension, which reduces the computation efficiency.

This defines the need for better optimisation frameworks and strategies to handle the high dimensional calibration optimisation problem while addressing the complex couplings between system control variables.

Lifeng *et al*. [215] employed a hybrid MDO approach for a four-cylinder gasoline engine to handle both *global* (i.e. minimise the fuel consumption over the drive cycle) and *local* (i.e. maximize torque at each minimap) objectives concurrently subjected to local constraints (i.e. exhaust temperature and cylinder pressure constraints), no global constraint were set for this formulation.

A similar approach was proposed by Yin [43] to implement a hybrid MDO framework (which is based on CO strategy), for optimally calibrating a complex diesel engine problem. The global objective of this formulation was to minimise the overall fuel consumption, subject to emission (i.e. NOx and Particulate Mass) and gradient constraints (i.e. maximum allowable range of change for calibration parameters by moving among minimaps). The local objective for this problem was to meet the requirements for noise and exhaust temperature at each minimap point. Applying a MDO approach for this problem enabled delivering a global calibration setting by optimising both levels of optimisation simultaneously, both global and local levels, without any need for a *posteriori* analysis (e.g. analysis of Pareto trade-off solutions). Yin [43] showed that MDO approaches are quite flexible frameworks for complex systems and can be efficiently implemented for engine calibration problems.

A summary of reviewed papers in this section, regarding application of various optimisation algorithms for engine calibration optimisation problems, is presented in Table 3.7.

Reference	Optimisation Approach	Optimisation Process
Millo *et al.* [211]	Single Objective Optimisation approach/ five different algorithms are also applied and compared: Brent Method, Discrete-grid bisection, LSGRG, NLPQL and Hooke-Jeeves Direct Search.	Maximising brake torque, subject to knock and brake torque constraints.
Sheridan *et al.* [59]	Single Objective Optimisation approach/Gradient descent algorithm.	Maximize Torque, subject to exhaust temperature and engine stability constraints.
Popovic *et al.* [212]	Single Objective Optimisation approach/different Gradient-based algorithms such as Box and Wilson Steepest Descent (BWSD).	Minimize BSFC, subject to exhaust temperature and engine stability.
Morton *et al.* [24]	Multi-Objective Optimisation approach/ NBI algorithm.	Minimise BSFC and NOx, subject to manifold pressure and exhaust temperature constraints.
Vossoughi and Rezazadeh [23]	Multi-Objective Optimisation approach/ two types of multi-objective genetic algorithms (NSGA and DPGA).	Minimise BSFC and NOx, subject to manifold pressure and exhaust temperature constraints.
Kaji and Kita [21]	Multi-Objective Optimisation approach/ NSGA-2 algorithm.	Minimise fuel consumption and maximise torque, subject to combustion stability.
Kaji *et al.* [213]	Multi-Objective Optimisation approach/ NSGA-2 algorithm.	This approach aims to select the optimum solution for input parameters, such as ignition timing and fuel injection timing, by studying the 'HC-NOx' trade-off.
Singh *et al.* [2]	Multi-Objective Optimisation approach/ NSGA-2 algorithm.	Maximise Torque and minimise SDNMEP, subject to SDNMEP constraints.
Seabrook *et al.* [22]	Multi-Objective Optimisation approach	Studying fuel consumption trade-offs versus combustion stability, and also load, at each minimap to select the best global solution.
Lifeng and Yunqing [215]	Multi-Disciplinary Optimisation approach.	A hybrid MDO approach is applied to minimise the overall fuel consumption and maximise torque at each operating point.
Yin [43]	Multi-Disciplinary Optimisation approach/ Standard GA and fmincon algorithms.	A hybrid MDO approach is implemented to minimise the overall fuel consumption, subject to several constraints.

Table 3.7: Summary of reviewed literature on applying optimisation techniques for steady state calibration problems

3.7 Summary

In this chapter, some of the frequently used optimisation strategies are reviewed within three main categories:

- Single-objective optimisation strategies, which have been developed to solve the optimisation problems with a single objective, such as classical algorithms (e.g. gradient-based, direct search) and evolutionary algorithms (e.g. GA and PSO). The efficiency of different single-objective optimisation algorithms are compared in Table 3.1. According to the existing literature, the evolutionary algorithms can significantly increase the chance of finding the global optimum solution, however; they are computationally more expensive than the classical methods.
- Multi-objective optimisation strategies, which have been developed to solve the optimisation problems with multiple objectives by providing the Pareto frontier, such as classical methods (e.g. one-at-a-time and simultaneous techniques) and evolutionary algorithms (e.g. NSGA-II). As illustrated in Table 3.3, among the reviewed multi-objective optimisation strategies NSGA-II has the ability to provide a uniformly distributed Pareto frontier independent from the function continuity.
- Multidisciplinary design optimisation strategies, which have been developed to solve the complex optimisation problems through architectural decomposition of the problem into a number of sub-optimisation problems (i.e. disciplines), while addressing the coupling among the variables. The MDO strategies are studied within two main categories, single-level (e.g. SAND, IDF and MDF) and multi-level (e.g. CSSO, BLISS, CO and ATC) MDO strategies. The efficiency of different MDO approaches are compared in Table 3.5 and Table 3.6. Table 3.5 shows that among the single-level MDO approaches only IDF has the potential to decompose an optimisation problem into a hierarchical structure, while SAND and MDF architectures have an absolute non-hierarchical nature. Also, based on the reviewed literature summarised in Table 3.6, ATC has shown a highly robust convergence behaviour for complex problems. Moreover, only ATC has the

ability to decompose an optimisation problem into more than two optimisation levels.

The existing literature on application of optimisation strategies to solve the steady state engine calibration problems is also reviewed in this chapter. Based on the review, although the conventional calibration optimisation strategies, which are mainly based on single-objective and multi-objective optimisation strategies, have the potential to find an optimum solution, they cannot guarantee to find the global optimum solution and the optimisation process is often iterative and time consuming.

Chapter 4: Research Methodology

4.1 Introduction

The main aim of this research is to enhance the steady state engine mapping and calibration procedure and validate it through case studies. The purpose of this chapter is to define the methodology for research based on the critical review of the current state in literature and a deeper review of the problem. The organisation of the chapter is as follows:

- Outline the GDI engine case study.
- Describe the research methodology taken to conduct the steady state engine mapping and calibration process for the GDI engine, including: design of experiment, experimental set-up, response surface modelling, and calibration optimisation.
- Outline the diesel engine case study and the corresponding research objectives.
- Provide the list of software packages and toolboxes used to conduct the research.
- Summarise the research plan of the following chapters.

4.2 GDI Engine Case Study

4.2.1 Analysis of GDI Engine Case Study

The GDI engine case study focused on the hot steady state mapping and calibration of a 5-litre naturally aspirated V8 Gasoline Direct Injection (GDI) engine with multiple injections ECU capability (i.e. AJ133 Jaguar engine).

The methodology adopted to conduct the GDI engine mapping and calibration process was based on the steady state model based calibration process [1]–[8], discussed in Section 2.2, as illustrated in Figure 4.1. This process starts with an analysis of the engine control parameters and responses of interest at each engine speed / load point, followed by planning DoE experiments, steady state

data collection on the engine dynamometer, fitting and validating engine models based on the collected data, and implementing an optimisation framework using the fitted response models.

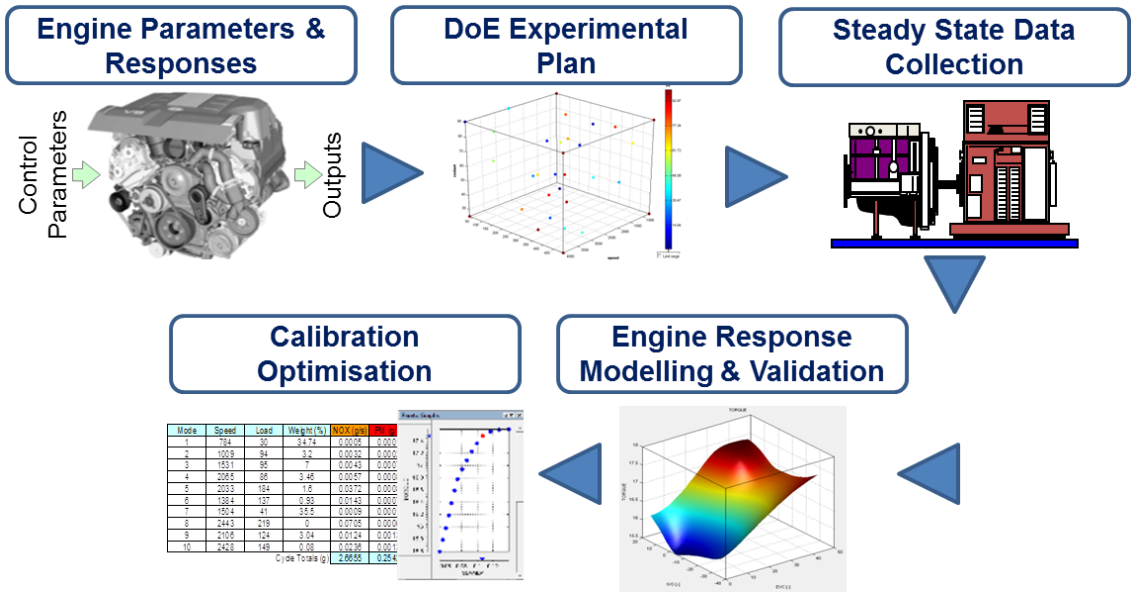


Figure 4.1: Steady state Model Based Calibration process applied for the GDI engine

A preliminary step is to establish the engine speed – engine load points (often called minimaps) for the steady state engine mapping. For the purpose of this study, which had a primary objective of developing a detailed characterisation of factors affecting the particulate number emissions, steady state engine tests were limited to 6 engine speed (rpm) / engine load (Nm) points, detailed in Table 4.1. These minimap points were selected by the sponsoring company to be representative of the part load operation of the engine within the testing capability of the facility. Figure 4.2 shows the full set of steady state minimaps used for the steady state calibration; the circles illustrate the 6 steady state minimaps selected for this study, detailed in Table 4.1. The weights (w_i) in Table 4.1 indicate the equivalent residency time for each minimap point, expressed as a proportion of the overall NEDC drive cycle time of 1186 seconds [20], for a target vehicle application of the engine. Figure 4.3 shows the AJ133 GDI engine cloud of residency points, representing the second-by-second predicted speed / load engine operating conditions over the NEDC drive cycle using three different vehicle applications (i.e. ‘X250 10MY AJ133NA’, ‘X250 13MY AJ133NA’, and ‘L405 13MY AJ133NA’). The weights given in Table 4.1 are

associated with the 'X250 10MY AJ133NA' application, which was proposed by the sponsoring company. These weights (w_i) have been calculated by adding up the seconds that engine spends at the residency points associated with nearest minimap point over the 1186 seconds of NEDC test duration.

Test Point (minimap)	Engine Speed (rpm)	Engine Load (Nm)	Weight (%)
1	700	28	0.47
2	1500	41	0.31
3	1250	125	0.09
4	1500	105	0.07
5	2000	81	0.03
6	2000	199	0.02

Table 4.1: AJ133 GDI engine minimap points

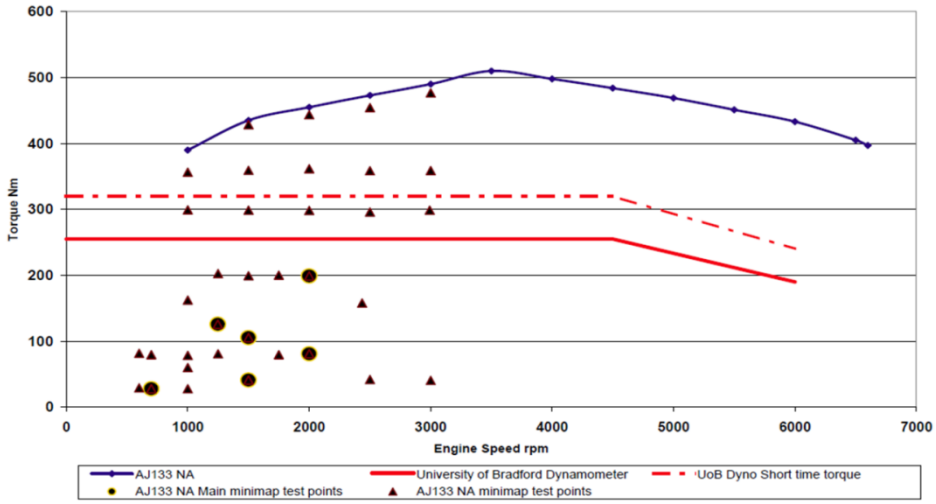


Figure 4.2: AJ133 GDI engine operating conditions within speed / load coordination (provided by Jaguar Land Rover)

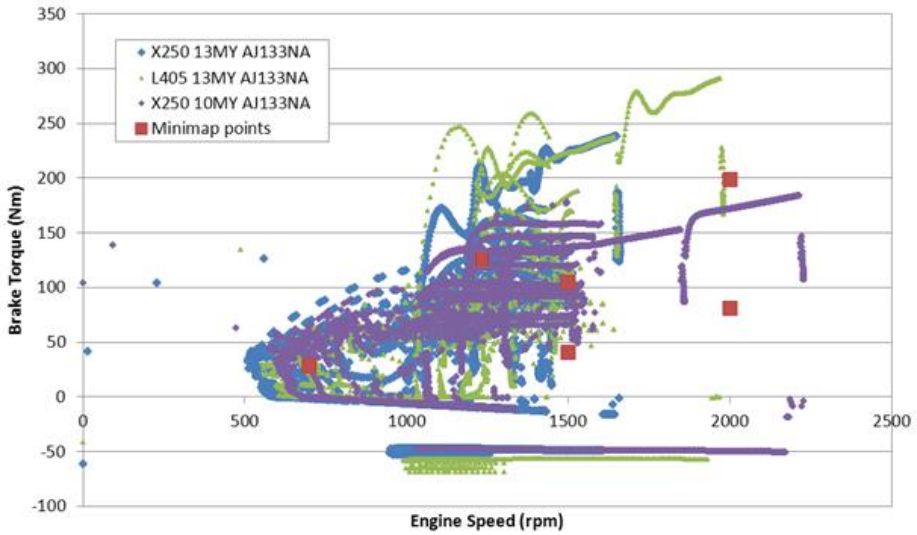


Figure 4.3: AJ133 GDI engine residency points during a NEDC test, for 3 different gearbox simulations (provided by Jaguar Land Rover)

4.2.2 Calibration Parameters and Responses

The GDI engine controllable calibration parameters of interest include:

1) **Valve timing:** Calibration of both Inlet Valve Opening (IVO) and Exhaust Valve Closing (EVC) parameters affects the mixture quality through controlling the air path, which provides fresh air and charge motion, and consequently affects the fuel atomization and evaporation [46]. Also, valve overlap, which is the time that both intake and exhaust valves are open (i.e. controlled by IVO and EVC), can affect the combustion stability. The effects of valve overlap are significantly based on the generated wave pressures at each steady state speed / load operating point [216]. In an ideal overlap situation, the fresh air replaces the exhaust gasses in the cylinder without passing through the exhaust system, which allows exceed fresh air in the cylinder. However, overlap is not efficient for all engine speed / load operating points. Generally, overlap is beneficial to torque at engine high speed and loads, whereas; it can have significant negative effects on emissions at lower speeds [217], [218].

2) **Fuel Rail Pressure (FRP):** Calibration of fuel rail pressure affects the fuel atomization and vaporization process. In general, increasing fuel pressure results in better fuel mixture quality due to better fuel atomization and evaporation, which consequently reduces the Pn. However, very high pressure fuel injection can cause the wall wetting problem [46].

3) **Spark Timing (ST):** The spark timing parameter can affect the emissions and fuel consumption by controlling the combustion timing. At a given speed / load operating point, retarding spark timing can lead to emission reduction due to late combustion and high exhaust gas temperature. However, retarded ignition can result in a penalty for fuel consumption [46].

4) **Start of Injection (SOI):** The start of injection parameter influences the engine responses significantly (e.g. emissions and fuel consumption) by controlling the available time for fuel to be atomised and evaporated before the combustion stage, which directly affects the fuel mixture quality [216]. Generally, for a GDI engine with single-injection strategy late injection is preferred at partial load operating conditions, which results in a stratified

charge by injecting the fuel during the combustion stroke. However, at higher load minimap points, early injection is desired which causes homogeneous charge by injecting fuel during the intake stroke [219].

Multiple injection events (i.e. double injections) were also considered for some of the engine speed / load operating points, since it was expected that using a multiple injection strategy for GDI engines would reduce particulate number (Pn) emissions and improve the fuel economy [46]. Accordingly, the fuel injection process using single injection strategy is controlled by 1 parameter, start of injection (SOI). However, for using double injection strategy, the injection process is controlled through 3 calibration parameters: Start of the first injection (SOI1), the delay between injection 1 and injection 2 (Delay), and proportion of total fuel injected in injection 1 (Ratio). The GDI engine calibration parameters and their engineering units are summarised in Table 4.2.

Code	***Name***	***Unit***
IVO	Inlet Valve Opening	deg ATDC
EVC	Exhaust Valve Closing	deg ATDC
FRP	Fuel Rail Pressure	MPa
ST	Spark Timing	deg ATDC
SOI1	Start of Injection 1	deg BTDC
Delay	Delay between injection 1 and injection 2	ms
Ratio	Proportion of total fuel injected in injection 1	%

Table 4.2: Engine control variables for the GDI engine case study

The GDI engine responses of interest with their engineering units are listed in Table 4.3. In this table, fuel consumption (FC) and emissions (Pn, HC, CO, CO_2 and NOx) are called *global responses* with the sense that calibration engineers are interested in the cumulative values of these responses over the drive cycle (i.e. NEDC), which is calculated for each global response *F* using Equation 4.1. In this equation, w_i denotes the residency weight for each minimap point as given in Table 4.1.

$$F = \sum_{i=1}^{11} w_i \times F_i \qquad \text{Equation 4.1}$$

Calibration engineers are also concerned about the *local responses* at each minimap point, which are important to deliver a stable combustion process. In Table 4.3, CoV(IMEP) and sdNMEP responses denote the combustion stability

at each minimap point . Based on the specified requirements by the sponsoring company, the combustion stability threshold for minimap points with Net Mean Effective Pressure (NMEP) more than 3 bar is set to 'CoV(IMEP) < 3%' and for minimap points with NMEP less than 3 bar is set to 'sdNMEP < 0.1 bar'. Moreover, the maximum allowable exhaust temperature (T_{ext}) at each minimap point is limited to 800 ^{o}C.

Engine Response of Interest	Coded Name	Unit
Fuel Consumption	FC	kg/hr
Particulate Number	Pn	p/ccm
Exhaust gas HC	HC	ppm
Exhaust gas CO	CO	vol %
Exhaust gas CO_2	CO_2	vol %
Exhaust gas NOx	NOx	vol %
Exhaust Temperature	T_{ext}	^{o}C
Coefficient of Variation of the Indicated Mean Effective Pressure	CoV(IMEP)	%
Standard Deviation of the Net Mean Effective Pressure	sdNMEP	bar

Table 4.3: GDI engine responses

Figure 4.4 summarises the GDI engine calibration parameters associated with the ECU control strategy, and main responses of interest for calibration optimisation at each minimap point.

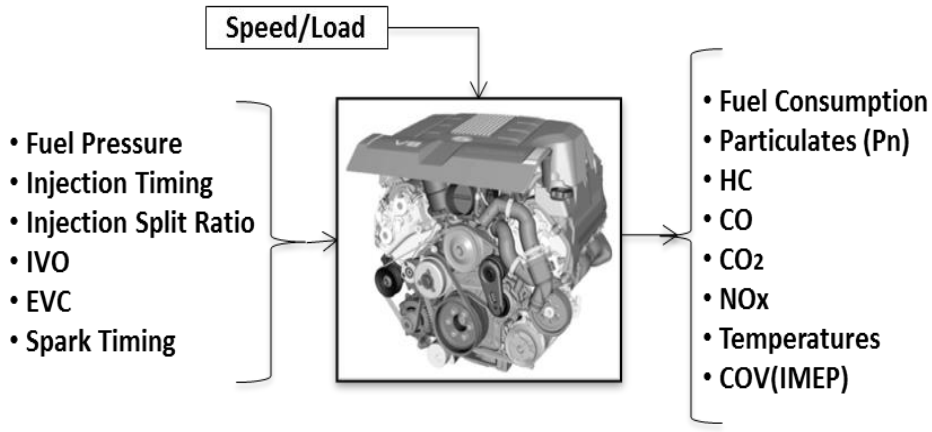


Figure 4.4: GDI engine control parameters and responses

4.2.3 Engine Mapping DoEs

There was a requirement to develop a DoE strategy which not only enables a wide exploration of the variables space (i.e. to define the effective calibration

parameters on the key engine responses and determine the final design space for each calibration parameter) but also facilitates a higher testing efficiency compared to the commonly used “one-shot” DoE strategies in practice [22], [24], [58], [73]–[75], [86], [136], [220]. The review of available literature on application of “one-shot” DoE strategies for the steady state engine mapping problems showed that there are some limitations to use these standard DoE strategies. Some of the limitations are the requirements of having prior knowledge regarding: the type of approximation models to represent the engine responses, the final design space for each calibration parameter (i.e. for Classical and Optimal DoEs), and the total number of required test points (i.e. for Classical, Optimal and Space filling DoEs).

Sequential DoE strategies have been introduced in literature as efficient DoE strategies for the engineering applications when not enough information regarding the system is available (e.g. model type, design space region of interest, and total number of test points) [96], [100], [105], [106]. An efficient sequential DoE strategy has the potential to address the ‘one-shot’ DoE limitations by flexibly adapting to the requirements of modelling complexity of responses, through the sequential modification of the experimental design [96], [100], [105], [106].

Accordingly, the research objective of enhancing the steady state engine mapping process is to develop a novel sequential DoE strategy, i.e. called Model Building – Model Validation (MB-MV), and validate it on the GDI engine case study. This exploration based sequential DoE strategy is designed based on using batches of OLH space filling DoEs [9], [11], [94] as the basis for subsequent DoE modifications, until the model accuracy is satisfactory. The reason of using OLH designs is the unique property of this DoE strategy to evenly cover the whole range of each design variable, as discussed in Section 2.3.3.

4.2.4 Engine Testing

4.2.4.1 Experimental Setup for Engine Dynamometer Testing

The engine set-up for the GDI engine case study at powertrain testing facility at the University of Bradford is shown in Figure 4.5 [221]. The GDI engine was

attached to a transient AC dynamometer. The steady state control method chosen was ‘speed-torque’, whereby the chosen engine speed would be maintained at the desired load by adjusting throttle position. Considering that the GDI engine case study was based on the ‘hot’ cycle of the GDI engine calibration problem, engine coolant was maintained at 90°C through external conditioning. Fuel was supplied to the high pressure fuel pump at 3 bar and 20°C. Ambient humidity, temperature and atmospheric pressure were all monitored but not controlled. A sampling rate of 1Hz was used and data was recorded for 60 seconds and time-synchronised between data sources.

Fuel consumption was measured using a fuel mass-flow meter that operates using a Coriolis sensor, which corrects for any density fluctuations. The engine was supplied with fuel from a loop that circulates via the fuel conditioner, which maintains the fuel temperature when there was low or zero flow. Gaseous emissions were sampled pre-catalyst from both banks in a 50/50 ratio and analysed using a HORIBA MEXA-9100 emissions bench. The particulate sample line was situated 3.2m downstream of the exhaust manifold (a pseudo tail-pipe position) and measurements were made using an AVL Condensation Particle Counter sampling raw exhaust gas. The instrument counts particles between 23nm and 2.5µm, i.e. the range specified in the proposed EURO VI legislation.

In-cylinder pressure measurement was available via an AVL Indi-Smart system and an ETAS-INCA interface with the engine ECU allowed engine parameters to be monitored and actuator settings to be controlled.

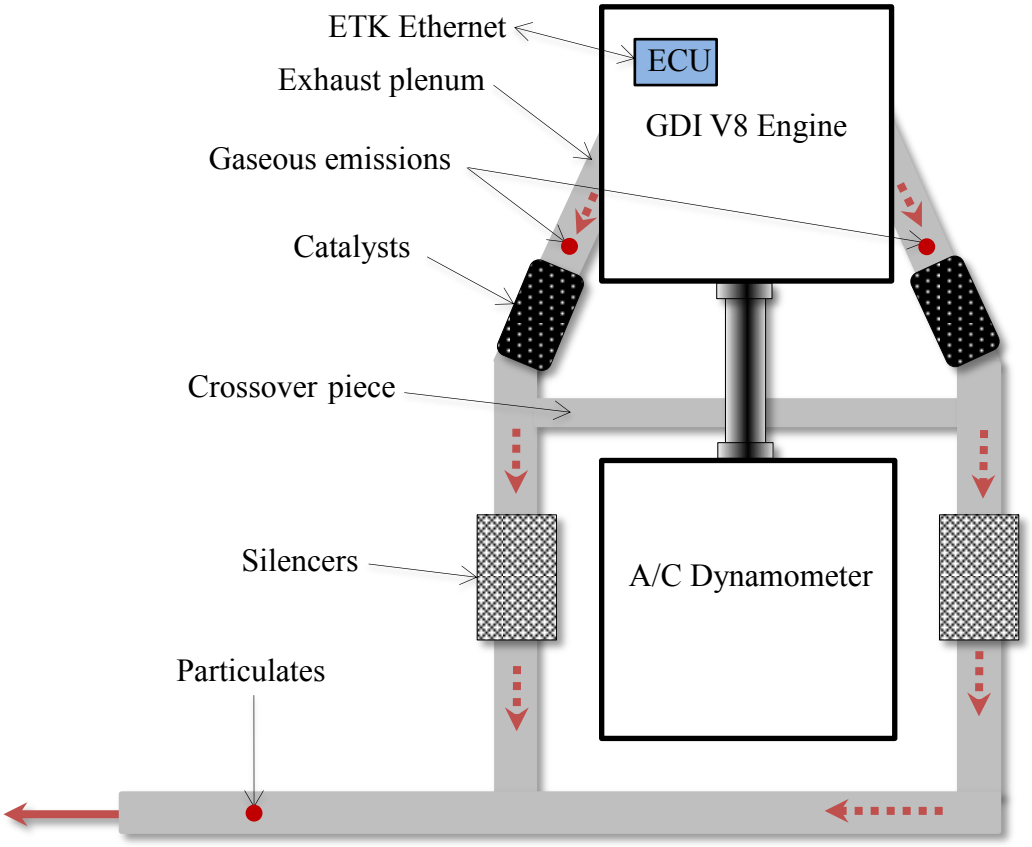


Figure 4.5: Schematic of the GDI engine test set-up at University of Bradford research centre

4.2.4.2 Steady State Engine Test Procedure

To acquire data that are representative of the chosen test configuration (actuator settings), the engine must be operating in a steady state condition. For the purposes of this investigation verification of steady state was achieved through the monitoring of certain critical temperatures. Engine coolant and oil temperatures change following a modification to operating conditions (speed and/or load). Engine oil is regulated internally via a 20kW oil-water heat exchanger, thus stabilizes shortly after the coolant temperature settles. Oil viscosity and thus friction losses are kept constant during testing by maintaining these temperatures.

A third temperature, exhaust plenum skin, is monitored in addition to the engine fluids. This value is measured immediately downstream of the exhaust manifold and before the catalyst (see Figure 4.4). When the value from this thermocouple changes by less than 1% over sixty seconds, a steady state operating condition

is assumed. This additional thermocouple was chosen to help establish when the engine had reached steady state. Particulate emissions are susceptible to variations in conditions both during and post combustion [222] and can continue to change for a significant duration following a change in speed and load. Exhaust plenum skin temperature indicates when the exhaust gas temperatures are stable, thus ensuring data are representative of the operating conditions. By following this strict test protocol, accurate and properly representative data are recorded which maximises the data quality used for model building.

In order to ensure that high quality data is consistently collected, a process has been set up where each data collection event was preceded by a baseline test with steady state data collected at a specific test point. In line with current test practice [223] data from repeat measurements was evaluated in terms of the coefficient of variation (CoV), which is calculated as the ratio of the sample standard deviation to the sample mean. The benefit of using CoV is that parameters of different orders can be compared and monitored during testing. This process was integrated with the routine test bed checks before any data were collected at steady state. Based on the analysis of the daily baseline tests, control charts could be established to ensure quality test data is consistently collected.

The collected data at each steady state operating condition was provided in Microsoft Office EXCEL format, containing values for responses of interest corresponding to the DoE test points. An example of the EXCEL data sheet is given in Appendix I.

4.2.5 Engine Response Modelling and Validation

Matlab Model Based Calibration (MBC) Toolbox™ [25] was used to characterise the engine responses of interest as a function of the calibration parameters, by fitting different types of statistical models (e.g. Polynomials and RBFs) to the collected data. A wide range of Radial Basis Function (RBF) models [121] with different Kernel functions were employed, due to ability of this modelling technique to learn the complex relationship between the parameters and responses [123], which would consequently result in fewer measurement requirements [131]. Moreover, considering the generalization capability of RBF

models to confront the missing and noisy data [123], these models were a suitable modelling candidate for the proposed DoE strategy (i.e. MB-MV DoE strategy).

The fitted approximation models were validated based on statistical criteria, e.g. PRESS-RMSE [224] and validation RMSE [156], as well as engineering judgement through analysis and validation of engineering trends [225].

4.2.6 Calibration Optimisation

4.2.6.1 Research Plan for Calibration Optimisation

The calibration study for the AJ133 GDI engine was particularly focused on developing an optimal calibration to meet forthcoming emission legislation (EURO VI) based on particulate number (Pn), while minimising fuel consumption.

The conventional 2-stage calibration optimisation approach, which is a common practice for the steady state engine calibration [2], [21]–[24], is based on solving the calibration problem within 2 independent optimisation levels (‘local’ and ‘global’ levels), as discussed in Section 3.6. It can be argued based on the available literature that the 2-stage approach cannot deliver the optimum calibration solution since it solves the two levels of optimisation separately. In this research, the objective of applying the conventional 2-stage calibration optimisation approach for the GDI engine case study, using the fitted engine response models at the mapping stage via applying the MB-MV DoE strategy, is to critically investigate advantages and disadvantages of this optimisation formulation.

Multidisciplinary Design Optimisation (MDO) frameworks have been introduced in literature as efficient optimisation techniques to deal with high dimensional optimisation problems with strong coupling interactions [27]–[33], as discussed in Chapter 3. These frameworks have the potential to address the shortcomings of the conventional 2-stage optimisation approach by solving the two levels of steady state calibration optimisation problem concurrently.

Accordingly, the research objective of the GDI engine calibration optimisation problem is to apply MDO optimisation methods to reformulate the steady state calibration optimisation problem in a way coherent with the 2-level structure of the model based calibration optimisation problem (i.e. both 'local' and 'global' levels). After reviewing the available literature on application of MDO frameworks to solve the complex engineering problems [27]–[34], [186], it was decided to apply 3 of the frequently used MDO optimisation architectures to formulate the GDI engine steady state calibration optimisation problem, as following:

- **Multidisciplinary Feasible (MDF):** MDF is a single-level non-hierarchical MDO approach [39] which is strongly capable of handling both local and global optimisation levels simultaneously [197]. The main reason for using the MDF optimisation strategy is the similarity of this optimisation technique to the optimisation approaches used by the currently available commercial calibration toolboxes in automotive industry, such as CAMEO Toolbox^{TM} [226].

- **Collaborative Optimisation (CO):** CO is a bi-level MDO approach which has been a popular optimisation technique for solving the optimisation problems with many coupling interactions among the parameters [33], [186], [197]. The main reason of applying the CO optimisation approach is that it has been frequently suggested in literature as a benchmark to investigate the performance of other MDO approaches [27]. A unique advantage of applying a CO framework is the fewer number of optimisation tune up parameters than the other hierarchical MDO approaches [33].

- **Analytical Target Cascading (ATC):** ATC is a multi-level MDO approach. The main motivation to apply the ATC optimisation approach is that ATC was originally developed for automotive industry (i.e. engineering product development area [42]). The main advantage of ATC approach compared to the other hierarchical MDO approaches is the robust convergence behaviour of this approach for large-scale optimisation problems.

Furthermore, given that double injection events was considered for some of the GDI engine minimap points, the model based calibration process for this case study (see Figure 4.1) is conducted for two independent calibration cases:

- **Calibration problem 'Case 1'**: GDI engine with *Single Injection (SI)* strategy;

  For this calibration problem, the steady state model based calibration process is conducted over the 6 speed / load operating conditions shown in Table 4.1 using the single injection strategy, as illustrated in Table 4.4.

- **Calibration problem 'Case 2'**: GDI engine with *Double Injection (DI)* strategy;

  In this case of calibration, the steady state model based calibration process is conducted over minimap points 1 and 2 using the single injection strategy (similar to calibration case 1) and minimap points 3 to 6 using the double injection strategy, as illustrated in Table 4.4. For minimap points 1 and 2 the amount of fuel injection at the steady state speed / load operating conditions was not enough to have more than a single injection.

Case \ Minimap	1	2	3	4	5	6
Case 1	SI	SI	SI	SI	SI	SI
Case 2	SI	SI	DI	DI	DI	DI

Table 4.4: GDI engine Calibration cases using different injection strategies

4.2.6.2 Benchmark Calibration Solution for the GDI Engine Case Study

The current calibration for the GDI engine case study, which was set by the sponsoring company, was used as the benchmark solution. Table 4.5 illustrates the benchmark calibration setting for the variables at each minimap point, and the corresponding response values for FC and Pn. Also, the overall FC and Pn values which are calculated using Equation 4.1 are shown in this table.

Speed/Load (rpm – Nm)	IVO (ATDC)	EVC (ATDC)	FRP (MPa)	SOI (BTDC)	Weight (%)	FC (kg/hr)	Pn (p/ccm)
700-28	50	-6	3	290	0.47	1.56	8.74E+04
1500-41	50	8	5	290	0.31	3.56	7.86E+05
1250-125	50	37	9	305	0.09	4.75	1.08E+06
1500-105	50	33	11	306	0.07	5.12	1.39E+06
2000-81	50	30	11	300	0.03	5.89	1.37E+06
2000-199	50	43	14	293	0.02	10.73	1.68E+06
					Overall	2.94	5.38E+05

Table 4.5: Baseline calibration solution for the GDI engine, set by JLR

4.3 Diesel Engine Case Study

The main objective of the diesel engine case study is to validate the applicability of using MDO architectures (i.e. MDF, CO and ATC) to solve the steady state calibration optimisation problems. One of the shortcomings of the GDI engine case study is that it is not a full calibration problem, i.e. covering the full engine speed / load space, since it was tested within the testing capability of University of Bradford testing facility. Therefore, an alternative case study was provided by the sponsoring company, which is based on full calibration of a diesel engine. The main reason of using an alternative case study is the need of applying the MDO frameworks on a highly constrained optimisation problem to demonstrate the full potential of MDO frameworks [27]. For the diesel engine case study, all the required information (i.e. steady state data and engine response models) was provided by the sponsoring company.

4.3.1 Case Study Outline

The diesel engine case study is based on the engine dynamometer experiments conducted in Jaguar Land Rover (JLR) test cell for the hot steady state calibration of an EU6 passenger car diesel engine. Steady state dynamometer test data was collected at 11 minimap points, which were selected to be representative of the engine residency across a range of drive cycle for a target vehicle application. Table 4.6 illustrates the selected minimap points and the corresponding residency (w_i) in the NEDC emissions drive cycle.

Test Point (minimap *i*)	Engine Speed (rpm)	Engine Load (Nm)	Weight (w_i) (%)
1	1000	95	0.15
2	1250	48	0.18
3	1250	84	0.21
4	1250	190	0.07
5	1350	25	0.07
6	1500	72	0.08
7	1500	143	0.1
8	1500	238	0.01
9	1750	131	0.04
10	1750	190	0.06
11	2000	240	0.03

Table 4.6: Diesel engine minimap points

Figure 4.6 summarises the engine calibration parameters and responses for the Diesel engine. These parameters and responses are detailed in Table 4.7 and Table 4.8, respectively. The steady state data collection process was conducted at JLR using on-line testing facilities. CAMEO Toolbox™ was used to plan the DoE strategy based on the calibration parameters (Table 4.7) and fit the engine response models of interest (Table 4.8). Accordingly, an adaptive D-Optimal DoE design with additional space filling infill test points was planned to collect data from each of the 11 steady state minimap operating points. Using this DoE strategy, the infill points were added to the initial D-Optimal design once the Convex hull (or Convex envelope) [227] had been determined. Convex hull was used to define the minimal convex which contains all the feasible tested DoE data points, to be established as the final design space. The number of calibration variables used at each minimap point was different across the engine speed / load operating space, as shown in Table 4.9. Accordingly, the number of DoE points collected at each minimap point, listed in Table 4.9, was adjusted to reflect the number of variables used and the desired model accuracy [223]. Subsequently, response models were fitted for each engine response (Table 4.8) on the test data collected at each minimap point, using CAMEO Toolbox™. Cubic models were found to be adequate, and each model was validated based on both statistical indicators and engineering judgment.

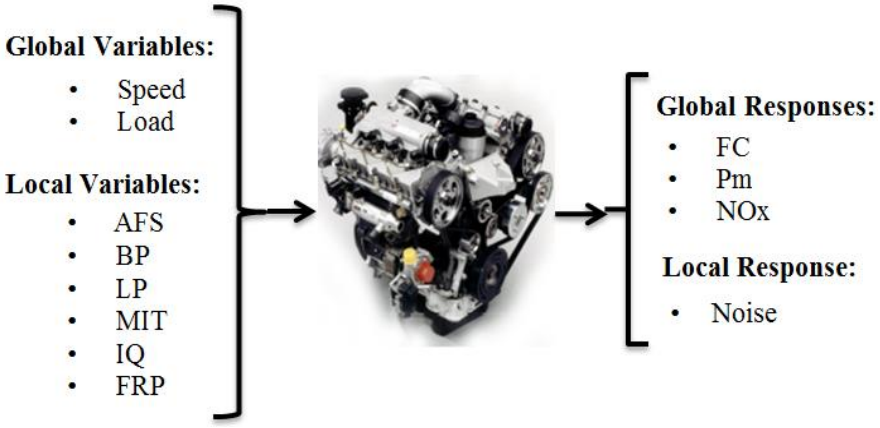


Figure 4.6: Diesel engine control parameters and responses

Calibration Parameter	Symbol	Unit
Air mass flow	AFS	mg/stroke
Boost pressure set point	BP	hPa
Low pressure EGR fraction	LP	%
Main injection timing	MIT	BTDC
Post injection quantity	IQ	mg/inj
Fuel rail pressure	FRP	hPa

Table 4.7: Engine control variables for the diesel engine

Calibration Responses	Symbol	Unit
Fuel consumption	FC	kg/hr
Particulate matter	Pm	gr/hr
NOx	NOx	gr/hr
Combustion Noise	Noise	dB

Table 4.8: Calibration responses for the diesel engine

Minimap	AFS	BP	LP	MIT	IQ	FRP	Test
1	✓	✓	✓	✓	✓	✓	128
2	✓	✓	✓	✓	NA	✓	117
3	✓	✓	✓	✓	✓	✓	80
4	✓	✓	✓	✓	NA	✓	125
5	✓	NA	✓	✓	NA	✓	82
6	✓	✓	✓	✓	✓	✓	127
7	✓	✓	✓	✓	✓	✓	127
8	✓	✓	✓	✓	NA	✓	122
9	✓	✓	✓	✓	NA	✓	170
10	✓	✓	✓	✓	NA	✓	116
11	✓	✓	✓	✓	NA	✓	66

Table 4.9: List of DoE parameters and sample size of each minimap point

Among the engine responses listed in Table 4.8, fuel consumption (FC) and emissions (Pm and NOx) are *global responses*, which are calculated over the

NEDC drive cycle using Equation 4.1 (i.e. w_i is given in Table 4.9). Moreover, the combustion noise (Noise) is a *local response* which was measured at each minimap point.

The calibration study for the diesel engine case study was focused on defining the optimal setting for the calibration parameters (i.e. 58 calibration parameters in total) to minimise the fuel consumption over the NEDC drive cycle, while satisfying the constraints for emission legislation (Table 4.10) and combustion noise (Table 4.11).

The upper limits for overall NOx and Pm (i.e. global responses) are illustrated in Table 4.10, which are set based on the engineering calibration targets for the hot calibration of the diesel engine. Also, the upper limits for combustion noise (i.e. local response) are set at each minimap point *i* (E_i), given in Table 4.11.

Constraint	**Limit**
Overall NOx (gr/hr)	6.9
Overall Pm (gr/hr)	3

Table 4.10: Global constraints (upper limits) over the NEDC drive cycle

Minimap	1	2	3	4	5	6	7	8	9	10	11
E_i	73	75.5	78	---	72.5	79	80	80	---	81	83

Table 4.11: Local constraint (upper limits) for Noise at each minimap point (E_i)

4.3 2 Benchmark Calibration Solution for the Diesel Engine Case Study

The benchmark solution for the diesel engine case study is the current engine calibration, which was set by the sponsoring company. This solution was generated by JLR calibration team using the AVL CAMEO optimisation toolbox. In this toolbox, the optimisation problem was formulated using a single-objective optimisation framework, aiming to minimise the fuel consumption over the NEDC drive cycle, subject to the constraints for emission legislation (Table 4.10) and combustion noise (Table 4.11). Accordingly, the problem formulation using the CAMEO toolbox is illustrated in Equation 4.2.

Objective:
Minimise $\sum_{i=1}^{11} w_i \times FC\ (X_i)$

With respect to
$X_i = (\mathrm{AFS}_i\ \mathrm{BP}_i\ \mathrm{LP}_i\ \mathrm{MIT}_i\ \mathrm{IQ}_i\ \mathrm{FRP}_i)$

Subject to:
$g_{s1} = \sum_{i=1}^{11} w_i \times Pm\ (X_i) \leq 3$
$g_{s2} = \sum_{i=1}^{11} w_i \times NOx\ (X_i) \leq 6.9$
$g_{s1i} = Noise(X_i) \leq E_i$
$g_{s2i} = C(X_i) \leq 0$
$X_i^{min} \leq X_i \leq X_i^{max} \qquad i = 1, \ldots, 11$ Equation 4.2

Where, *i* indicates the number of minimap points, w_i represents the equivalent residency time for each minimap point *i* at NEDC drive cycle as given in Table 4.6, g_{s1} and g_{s2} are the global constraints for overall Pm and NOx, respectively, g_{s1i} is the local constraint for the combustion noise upper limit (E_i) at each minimap point *i* as illustrated in Table 4.11, g_{s2i} is the convex hull constraint for each minimap point *i* which defines the feasible design space for each parameter, X_i denotes the calibration parameters at minimap point *i* as illustrated in Table 4.7, and X_i^{min} and X_i^{max} are the lower and upper boundary limits for the calibration parameters, respectively.

CAMEO software used a Hybrid Genetic Algorithm (GA) to solve the diesel engine calibration optimisation problem. Hybrid GA combines a global optimisation algorithm (GA) with a Sequential Quadratic Programming (SQP) algorithm, to increase the speed of the convergence. Considering the fact that the GA is a stochastic process and might converge to different optimum solutions for different initial populations, the optimisation was run 5 times to enhance the robustness of the optimisation framework.

The benchmark calibration setting for the diesel engine case study (i.e. CAMEO solution) is illustrated in Table 4.12. In this table, the calibration setting for all the 58 calibration parameters, and also the corresponding engine responses at each minimap point and over the NEDC drive cycle (i.e. calculated using Equation 4.1) are illustrated.

Minimap	AFS (mg/str)	BP (hpa)	LP (%)	MIT (BTDC)	IQ (mg/inj)	FRP (hpa)	FC (kg/hr)	Pm (gr/hr)	NOx (gr/hr)
1	286.8	1098.5	81.1	-3.2	1.7	6.30E+05	0.32	0.16	0.22
2	247.5	1053.6	70.7	0.4	NA	6.43E+05	0.28	0.03	0.50
3	301.3	1045.6	93.0	-0.3	2.2	5.71E+05	0.53	0.12	0.67
4	511.8	1413.5	99.7	-2.5	NA	1.03E+06	0.40	0.30	1.35
5	231.5	NA	5.5	-2.9	NA	4.72E+05	0.08	0.02	0.43
6	309.5	1103.8	87.7	-0.3	1.7	6.43E+05	0.20	0.03	0.24
7	441.6	1321.7	89.1	-4.1	2.1	7.07E+05	0.52	0.52	0.83
8	623.1	1682.5	100.0	-2.3	NA	9.43E+05	0.09	0.05	0.31
9	417.0	1317.4	80.7	-0.6	NA	8.18E+05	0.23	0.25	0.36
10	543.8	1617.7	98.9	-2.8	NA	8.75E+05	0.49	0.28	0.93
11	633.2	1748.2	91.8	-3.8	NA	1.25E+06	0.37	0.16	0.78
						Overall	3.51	1.92	6.62

Table 4.12: Optimum setting for the calibration parameters using CAMEO software (provided by JLR)

4.4 Software Tools

Different programming software and analysis tools were used during the course of studies, as detailed below:

- **Model Based Calibration Toolbox™ (MBC) version 4.1:**

  MBC [25] is a commercial package developed by *Mathworks* and *Ford Motor Company.* This package is providing tools (e.g. design of experiment and statistical modelling) to support calibration of engine systems. In this research, MBC toolbox was mainly employed for:

  - Fitting approximation models for the responses of interest. MBC provides an extensive range of modelling techniques, such as Polynomials and Neural Networks.
  - Adding boundary constraints to the approximation models. MBC lets the user to build a boundary constraint that can define the feasible regions of engine operating envelope. Boundary constrains results in avoiding selecting calibration solutions from the regions which cannot be physically tested.
  - Validating the fitted approximation models. Several statistical metrics are defined in MBC interface (e.g. Root Mean Square Error (RMSE)

and Prediction Error Sum of Squares (PRESS)) to validate the approximation models statistically.

- Visualisation of fitted models. The visualisation tools in MBC provide the opportunity to study behaviour of fitted responses over the parameters range, and determine any possible outliers to improve the model quality.

- **AVL CAMEO™ Toolbox version 2013 R2:**
  CAMEO [226] is a commercial powertrain calibration software developed by AVL. Similar to MBC software, this toolbox contains various design toolboxes (e.g. design of experiment, response modelling and optimisation) which empowers the calibration engineer to execute the complete calibration process. This software can be used either 'off-line', on a personal computer, or 'on-line', connected to the test bed, based on the available testing facilities. In this research, CAMEO toolbox was used 'off-line' to analyse the diesel engine models and calibration solutions provided by the sponsoring company.

- **Minitab® Toolbox version 16:**
  Minitab® is a commercial statistics package developed by Pennsylvania State University. This package was mainly used to analyse the statistical properties of the MB-MV DoE strategy.

- **MATLAB® programming environment version R2010b:**
  MATLAB® is a programming language developed by *Mathworks*. Given that both MBC and CAMEO software packages can be used within the MATLAB environment and also many optimisation algorithms are available in MATLAB Global Optimisation Toolbox [228], hence; MATLAB® was used as an ideal environment to program the entire steady state engine calibration optimisation problem.

4.5 Implementation plan

Figure 4.7 summarises the research methodology used to conduct this research over the following chapters.

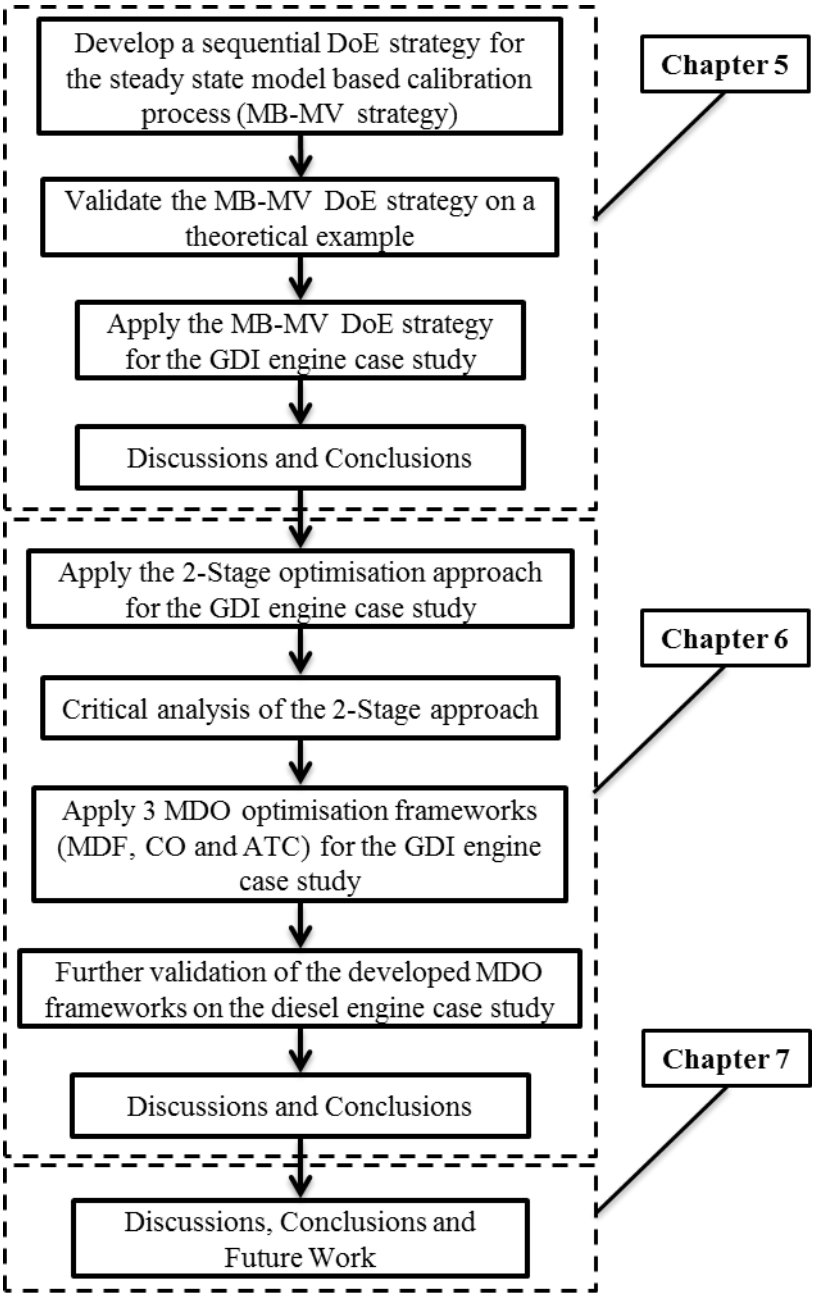


Figure 4.7: Research implementation plan

Chapter 5: Development of a Sequential DoE Strategy for the Steady State Model Based Calibration Process

5.1 Introduction

This chapter presents research to improve model based engine calibration methods using an integrated sequential DoE strategy, in conjunction with the GDI engine case study. The proposed DoE framework combines a Model Building space filling DoE, principally aimed to model the engine responses based on the effective parameters, with a flexible sequence of model validation DoEs. The process used to carry out this investigation consisted of the following steps:

1. Highlight the main shortcomings of existing one-shot DoE approaches for steady state engine mapping and describe the research motivations to develop a novel DoE strategy.
2. Propose a sequential exploration-based DoE strategy, called Model Building - Model Validation (MB-MV), and outline the main principals.
3. Implement the sequential MB-MV DoE strategy using an evolutionary optimisation algorithm, i.e. Permutation Genetic Algorithm (PermGA).
4. Validate the MB-MV method through a preliminary mathematical problem, i.e. six-hump-camel-back function which is a theoretical function proposed in literature [229].
5. Apply the MB-MV strategy to a series of mapping experiments for the GDI engine case study, consisting of screening, model building, and model validation sequences.
6. Comprehensive discussion of the results and making conclusions.

5.2 Research Motivation

In general, an efficient DoE strategy aims to minimise the cost of testing while maximising the information content, such that response surface models of

specified approximation accuracy can be developed [9]. The practical importance of choosing an efficient DoE strategy is associated with the high cost of engine testing. To select an efficient DoE strategy for an engine model based calibration problem in practice, there are several decisions to be made in advance [101], such as ‘what is the best statistical approximation technique to model the engine response behaviour?’, ‘what is the best DoE technique?’, and ‘how many overhead test measurements are required to handle the measurement and system noise?’. For many engine calibration cases where not enough information regarding the engine behaviour and influence of calibration parameters on the responses of interest are available, answering to these questions is a challenging task for the calibration engineers.

Given that modern engine calibration problems involve an increasing number of calibration variables, conventional Classical DoE strategies (such as factorial based DoEs) are generally not feasible or economical. A further complication of engine experiments is that the design space is often not orthogonal with respect to one or more variables (i.e. linear or nonlinear constraints limit the actuator space in one or more dimensions), which further limits the applicability of both Classical and Optimal DoEs.

The literature review showed that the D-Optimal, V-Optimal and space filling DoEs are the most commonly used DoE methods for engine mapping experiments [73]–[75], [86]. Optimal DoEs require prior knowledge regarding the model type to guarantee an efficient experimental design, consequently; Space filling DoEs, in particular based on Optimal Latin Hypercubes (OLH), have been increasingly used for engine model based calibration problems. OLH designs enable more flexible models to describe the engine behaviour over a wider design space (e.g. ‘global’ models over the whole engine speed / load space) [102], with no prior knowledge required regarding the type of model that would adequately represent the trends [22]. The OLH DoEs have the advantage that the number of test points can be set by the analyst, based on experience and resource limitations (unlike the Classical designs that the total number of DoE tests is not flexible [71]). However, this raises the risk of test plans that generate an insufficient amount of information due to *under-sampling*, with the implication that the required model accuracy is not achieved. Conversely, if a

larger OLH DoE test plan is selected, this raises the possibility of *over-sampling*, wasting time and energy by collecting more tests than needed.

Consequently, applying a sequential DoE strategy can be a potential solution to address shortcomings of currently used single-level DoE strategies for model based calibration process. The idea of sequential DoE approaches is to iteratively augment an initial DoE with further test points until the desired model quality is reached [96], [100], [105], [106]. This strategy can facilitate a higher testing efficiency compared to the fixed size tests commonly used in practice, and has the advantage that it can flexibly adapt to modelling complexity requirements of different engine responses.

Given that engine calibration task is a multi-objective optimisation problem with conflicting responses involved, using an exploitation-based sequential DoE [100], [108] may result in focusing on a specific part of design space that may not cover the optimum area for all the engine responses. This means that some of the optimisation solutions can be dismissed due to an impractical DoE strategy. Therefore, an exploration-based sequential DoE [65], [108], [109] can be a better solution to generate an experimental plan for an unknown engine model based calibration problem. An example of a sequential strategy used in engine testing was presented in [106], based upon a Sobol sequence method [115]. This approach has the shortcoming that the Sobol sequence is not adaptive with respect to 'learning' from previous stages and the selection of new test points is quasi random from of any taken subset of design space where the discrepancy is low, as discussed in Section 2.3.2.1.

Therefore, the strategy adopted in this research was to develop an exploration-based sequential DoE method, called 'Model Building – Model Validation' (MB-MV) design, which uses OLH space filling DoEs [9], [11], [94] as the basis for subsequent DoEs.

5.3 OLH Based Sequential DoE Strategy for Engine Mapping Experiments

5.3.1 Model Building – Model Validation (MB-MV) Strategy

As discussed in Chapter 2, the Latin Hypercube (LH) principle is to divide the space on each dimension into *n* (the number of test points) equal intervals, with one test point collected in each interval. LH based DoEs are popular space-filling DoE techniques due to their unique ability to generate *non-collapsing* designs, which is essential in ensuring uniformity of space exploration in all dimensions [3]. This is particularly important for multivariate problems where some variables might have only a minor influence on the response. The Optimal Latin Hypercube (OLH) DoEs are generated based on the LH principle by minimising a chosen metric for space filling or uniformity metric. Several uniformity metrics have been described in literature [10], [11], as previously reviewed in Chapter 2 (see Table 2.2).

In order to generate an optimal sequence of DoEs according to the MB-MV principle, it is proposed that the space filling criteria is maintained throughout the MB-MV sequence. What this means is that a MV DoE is generated as an OLH, but the optimality criterion defined in relation to the space filling metric for the overall DoE sequence (i.e. including all MB and MV DoE test points) is optimised. The joint DoE created by the union of the MB DoE (based on *m* levels) and MV DoE (based on *v* levels) test points will not strictly fulfil the LH principle as it is based on *m* + *v* interlaced and, in general, unevenly distributed levels.

This proposed strategy provides a good fit with the practical requirements of the steady state engine testing problem. According to this framework, a smaller model building (MB) OLH DoE experiment can be planned (e.g. *m* =50 test points), followed by a validation (MV) DoE experiment (e.g. *v* =15 test points). The MV is also an OLH (defined over 15 levels), but the optimality criterion is to minimise the space fillingness metric across the union of the MB and MV sets (*m* + *v* = 65 test points). Engine response models are fitted based on the MB DoE data, typically using non-parametric or semi-parametric models (such as Kriging or Radial Basis Function (RBF)) since a limited number of sample points are generated at each stage [96]. The quality of the models is typically

evaluated via the prediction error (e.g. using the Root Mean Squared Error, RMSE [156]) for the validation set , i.e. the MV DoE test data. Then, if the model accuracy requirements are not met, a further validation DoE test (MV2) is planned with v_2 test points, using the same principle of a OLH MV DoE in which the space fillingness metric is minimised across the MB + MV + MV2 set. A new model is fitted using the MB + MV set, and validated against the MV2 set based on RMSE. This process is repeated iteratively until the model accuracy requirement is met. The principle of the approach is illustrated as a flowchart in Figure 5.1.

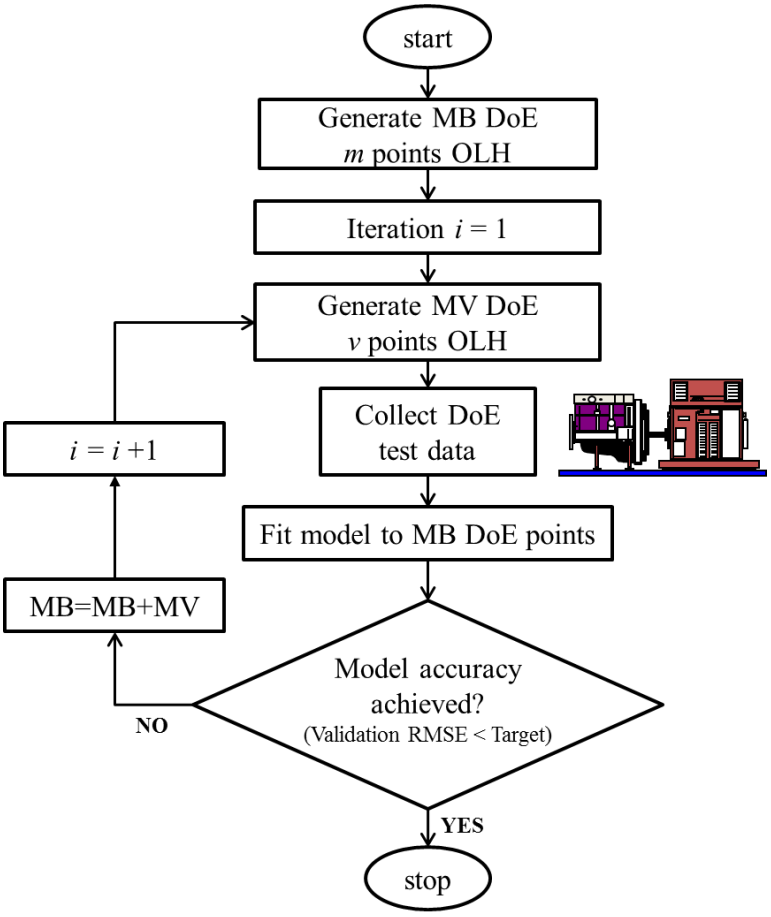


Figure 5.1: MB-MV Strategy Flowchart

5.3.2 Critical Analysis of MB-MV DoE Statistical Properties

The effectiveness of MB-MV strategy for a steady state engine mapping application can be assessed based on the essential statistical properties required to achieve a good sequential space filling DoE (i.e. space-fillingness, projection, orthogonality and granularity metrics), as discussed in Chapter 2. Noteworthy that the importance of these metrics for evaluating efficiency of a

DoE strategy varies based on the problem. For instance, a good projection property is vital for an experimental design to guarantee that the planned experiment can cover whole design domain for each parameter. However, avoidance from replication of projected points along one of the axes is mainly desirable for computer science problems (rather than engine applications) that are not subject to variation [76], [79], [230]. The performance of MB-MV DoE strategy for a steady state engine mapping application can be evaluated as following:

- **Projection property:** MB-MV design uses OLH DoEs as the basis for the subsequent DoEs due to their unique non-collapsing (i.e. good projection) property. Therefore, MB-MV design can ensure that whole range of design space for each parameter is covered by both individual designs (i.e. at each design sequence) and the final design (i.e. combination of all individuals).
- **Granularity property:** Unlike the other sequential DoE strategies based on OLH designs, such as the sequential Nested Latin Hypercube DoE method [65], MB-MV design is fine-grained. So, in this design the number of additional MV points at each sequence is arbitrary, e.g. small batches of 10 to 15 test points, whereas the nested LH design doubles the number of test points at each iteration.
- **Space-fillingness property:** To maintain a good space-filling design through several iterations of DoE merging, selecting an efficient space-filling metric (see Table 2.2) and a proper optimisation algorithm, explained in Chapter 2, plays a key role.
- **Orthogonality property:** An orthogonal DoE assures that there is no correlation between each combination of input parameters [82], [83], as explained in Chapter 2. In general, given that the vectors of calibration setting for each of the parameters are selected randomly, it is expected that the final design satisfies the orthogonality criterion by using an efficient optimisation algorithm.

Therefore, the main challenge of implementing the MB-MV DoE strategy is to develop an algorithm which can preserve both space-fillingness and

orthogonality requirements for a space-filling experimental design. It should be highlighted that considering the MB-MV design principles (i.e. based on OLH designs), the granularity and projection criteria are suitably satisfied for the engine mapping application.

5.4 MB-MV OLH Algorithm Development

The computation challenge is to develop an efficient algorithm to support the implementation of the proposed MB-MV strategy. The algorithm development and implementation for the proposed sequential DoE strategy involved three stages:

(1) Algorithm development for generating space filling OLH DoEs (for the initial MB DoE), which can generate OLH DoEs with good space filling properties.

(2) Algorithm development to generate the infill test points as OLH DoEs (for MV DoEs), which can generate an infill set of validation points as an OLH DoE that would project good space filling properties in conjunction with the initial DoE.

(3) Modification of the above algorithms to cope with nonlinearly constrained design space, thus; the algorithm must be robust to non-orthogonal variables design spaces.

5.4.1 OLH DoE Algorithm Implementation

The distribution of the test points for a OLH design is regarded as a discrete optimisation problem [10]. The main challenges with this optimisation problem are:

i. Formulation of the optimisation objective function to maintain space-fillingness.

ii. Development of an effective algorithm for the discrete optimisation problem.

5.4.1.1 Optimisation Objective Function

Both the model building and model validation DoEs are individual Latin Hypercube designs (i.e. satisfying the condition of one point per level [86]), generated based on optimising a space filling or a uniformity metric. Several uniformity metrics have been described in literature (see Table 2.2). While Maximin is a common objective function [80], it was shown that it tends to start the sample generation around the corners , especially for high dimensional problems, and then augment the design at the centre of design space [96]. Therefore, Maximin function might not preserve a good space filling properties for small number of sample points, which is the case during the successive MV stages. The metric chosen for this implementation is based on the Audze-Eglais (AELH) potential energy concept, which has shown good ability to distribute the points uniformly within the design space regardless of the sample size [11], [13].

The AELH function is described as a system consists of points of unit mass within design space, in which points are exerting repulsive force on each other, potential energy, until all the points reach their equilibrium position [9]. This system has its minimum potential energy in an equilibrium state. The AELH function is based on the fact that the magnitude of the repulsive forces is inversely proportional to the square of the distance between the points in the system [156]. Thus, the AELH objective function can be presented as follows [9], [11], [97]:

$$U = \sum_{p=1}^{P} \sum_{q=p+1}^{P} \frac{1}{{L_{pq}}^2} \qquad \text{Equation 5.1}$$

where U is the potential energy, P denotes the total number of DoE points, and L_{pq} is the distance between any 2 points p and q, $p \neq q$. Minimising the potential energy ensures a uniform distribution of the sample points within the design space.

5.4.1.2 Optimisation Algorithm

Different global optimisation algorithms have been proposed in literature for generating OLH designs, such as columnwise-pairwise, simulated annealing, and Permutation Genetic Algorithm (PermGA) [9]–[11], [97] (see Table 2.3). It has been argued in the literature that PermGA can generate better distributed

points for high dimensional DoEs while reducing the computational costs [10], [11]. Therefore, in this thesis, a PermGA algorithm was developed and implemented in MATLAB® programming environment to generate OLH designs.

The PermGA algorithm work flow is illustrated in Figure 5.2. The main difference between standard *GA* and *PermGA* algorithm is the way GA operators (i.e. crossover and mutation) are formulated [10], [11].

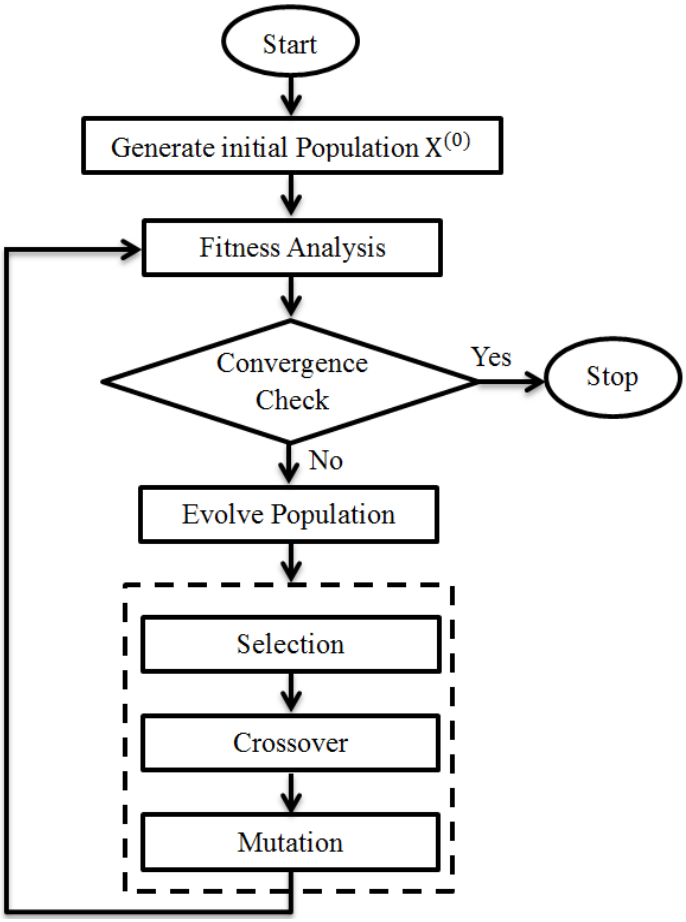


Figure 5.2: PermGA Flowchart

To enhance the exposition of developed PermGA algorithm, an extended design structure matrix (XDSM) [231] diagram was employed to visualize the interconnections among the PermGA components, as shown in Figure 5.3. XDSM is principally structured on the design structure matrix (DSM) diagram [232], which is a common practice in system engineering. DSM diagram is generally used to illustrate the components and the connections among them, however; this diagram does not demonstrate the meaning of the connections clearly (unlike the XDSM).

Using the XDSM architectural decomposition strategy [231], the PermGA components are represented by rectangles, the special components which control the iterations (known as drivers) are shown by rounded rectangles, and the data is represented by parallelograms. The function of components is to process data. The thick grey lines are used to show the data flow, while the thin black lines illustrate the process flow. The input data transfers to the components from the vertical direction and departs the components from the horizontal direction. The convention for the data flow is that connections above the diagonal flow from left to right and top to bottom, and connections below the diagonal flow from right to left and bottom to top. Accordingly, parallelograms at the column above and below the components define the input data, and parallelograms along the row define the output data. Moreover, external inputs and outputs are placed on the outer edges of the diagram, in the top row and leftmost column, respectively [231]. Another XDSM architecture convention is that any block referring to component *i* represents a repeated pattern. In addition, a numbering system is used to illustrate the order of components execution, i.e. it starts from zero and proceeds in numerical order. In this numbering system the loops are shown by $j \rightarrow k$ for $k < j$, which denotes that the algorithm returns to step *k* until the required termination condition by the driver is met.

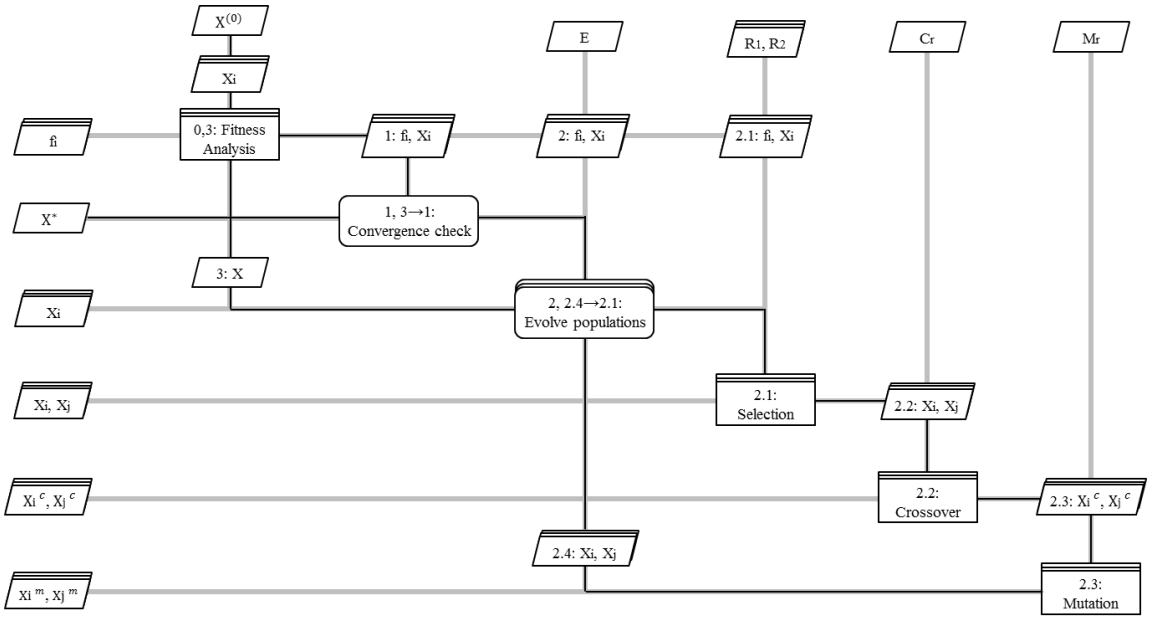


Figure 5.3: Illustration of PermGA using XDSM graph

For implementation, as shown in Figure 5.3, the external inputs at **level 0** (shown as **X**) are N populations of Random Latin Hypercube (RLH) designs which have been generated using the 'Permutation' encoding [158] in Matlab. Accordingly, the numbers in each string are represented in a sequence using integer numbers. For example, in order to generate two populations with 10 points, one of the possible solutions can be:

Population 1:	1 4 7 9 6 3 5 0 2 8
Population 2:	9 3 2 5 8 1 6 0 4 7

At **level 0** the optimisation objective function is calculated for each input population $\mathbf{X}_i$ ($i = 1, \dots, N$). This iterative analysis component is shown as 'Fitness Analysis' in Figure 5.3. The metric chosen for this implementation is based on the Audze-Eglais (AELH) given by Equation 5.1. Given that each DoE parameter might have a different range of units, the variables have each been normalised to the interval [0 1] in order to calculate the distance between any 2 points L_{pq}. The benefit of normalisation is that the sample points will be distributed in all the design dimensions without any preferences.

The fitness value for each population $\mathbf{f}_i$ and input populations $\mathbf{X}_i$ are transferred to **level 1** to check the algorithm convergence. If the convergence requirements are met, the optimum solution (X^*), which is the final OLH design, is delivered. Otherwise, the fitness value for each population $\mathbf{f}_i$ along with the input populations $\mathbf{X}_i$ will be transferred to **level 2** (shown as 'Evolve populations').

In **level 2**, GA operators are applied to the initial populations (called parents) to evolve them into new populations (called children). In this implementation, 'selection', 'crossover' and 'mutation' GA operators are utilised, shown as **level 2.1**, **level 2.2**, and **level 2.3**, respectively, in Figure 5.3.

- **'Selection' operator (Level 2.1)**: This operator defines the method of selecting the parent populations to be evolved. Several selection methods have been discussed in literature, including 'Tournament' and 'Biased Roulette Wheel' [159]. In this case the 'Biased Roulette Wheel' operator was implemented to increase the convergence rate, by giving individuals with better

fitness values $\mathbf{f}_i$ proportionally more chance to be selected as parents. R_1 and R_2 external inputs are two random numbers which define the parent populations ($\mathbf{X_i}$ and $\mathbf{X_j}$).

- **'Crossover' operator (Level 2.2)**: This combines parts of input parent populations (X_i and X_j) and generate two new individuals (X_i^c and X_j^c). There are different crossover methods used for PermGA, such as: simple crossover, cycle crossover and inversion [10].

a) Simple Crossover: In this operator, a crossover point is randomly selected and the permuted elements are copied from the first parent, the first element to the crossover point, to the child1. After that, the elements of the second parent are scanned and if the element is not copied yet in the child1, it will be added. The same procedure happens for the second parent. This process is shown in the followed example [10].

Parent 1 = [4 1 3 5 2] ⟹ Child 1 = [4 1 5 2 3]
+ +
Parent 2 = [5 2 1 4 3] ⟹ Child 2 = [5 2 4 1 3]

b) Cycle Crossover: In this technique, each element and its position in an individual string (child) are derived from one of the parents. As it is shown in the following example (Table 5.1), the position of elements in the parents sequence is preserved [10].

Parent 1: Parent 2:	[1 3 9 7 5 4 6 2 8] [4 6 2 1 7 8 9 3 5]
Child 1: taking the first element of parent1.	[1 * * * * * * * *]
Next point, value below1 in parent 2.	[1 * * * * 4 * * *]
Next point, value from parent 2 below the selected value from parent 1.	[1 * * * * 4 * * 8]
This rule continues until the cycle is complete and the next number is already used.	[1 * * 7 5 4 * * 8]
The remaining values are copied from parent 2. Similarly, the second child is generated.	[1 6 2 7 5 4 9 3 8] [4 3 9 1 7 8 6 2 5]

Table 5.1: An example of 'Cycle Crossover' technique

c) Inversion: In this method, two random elements of each parents is selected and the elements between these two points are inverted to generate the child;

Parent 1 = [1 3 9 |**7 5 4 6**| 2 8] ⟹ Child 1 = [1 3 9 |**6 4 5 7**| 2 8]

Bates [10] has shown that either cycle crossover or inversion work well for a PermGA algorithm. However, given that the interactions among GA parameters are complex and dependent on the fitness function [151], it was decided to employ both crossover functions, i.e. cycle crossover followed by inversion. The aim was to introduce extra variability into the children populations in order to reduce the chance of the search algorithm being trapped in a local optima.

- **Level 2.3**, **'Mutation' operator**: A simple mutation technique [11], [158] was used to swap two randomly selected elements of the transferred individuals, from the crossover level (X_i^c and X_j^c), and evolve them into new child populations (X_i^m and X_j^m). Following shows an example of Mutation where the position of the 2nd and the 5th elements of the vector are swapped.

Parent 1 = [4 *1* 5 2 *3*] ⟹ Child 1 = [4 *3* 5 2 *1*]

The output of the iterative process at level 2 is N new populations which are transferred to **level 3** to evaluate the fitness of evolved populations. The convergence requirements are checked again at level 1. If the convergence requirements are met, the optimum solution (X^*), which is the final OLH design, is delivered. Otherwise, the new population will be transferred to level 2 for another iteration of program.

Additionally, some PermGA parameters, shown as external inputs in Figure 5.3, require tuning due to their significant influence on the algorithm performance [162]; specifically:

- **Elite Size** (**E**): defines how many individuals with the best fitness value f_i (i.e. called elite individual) should transfer to the next iteration of the algorithm without evolving. Thus, the best individuals are not lost during subsequent generations, which can accordingly assure a smoother convergence [233].
- **Crossover Rate** ($\mathbf{C_r}$): determines the number of individuals that are evolved through the crossover operation.
- **Mutation Rate** ($\mathbf{M_r}$): determines the number of individuals that are evolved through the mutation operation.

- **Population Number** (**N**): denotes the population size for the input design X.

The pseudocode for the developed PermGA algorithm is listed in Table 5.2.

Input: Initial populations of RLH designs ($X^{(0)}$)
Output: Optimal Latin Hypercube design (X^{*})

0: Initiate the optimisation process
0: Evaluate the objective function for each population *i* (X_i)
for each population *i* do
Evaluate the fitness function (f_i)
end for
Until 3→1: Convergence criterion is fulfilled (i.e. maximum iteration)
Repeat
1: Convergence check
2: PermGA operators Analysis
Until 2.4→2.1: (*i* – E) new populations are generated, E of the populations with best fitness values are carried over without evolution to the next iteration (i.e. Elite).
Repeat
2.1: Select 2 populations as parents (using R_1 and R_2 random numbers)
2.2: Evolve the parents using the Crossover operator (based on crossover rate C_r)
2.3: Evolve the parents using the Mutation operator (based on crossover rate M_r)
2.4: Compute the new populations (i.e. children)
3: Evaluate the objective function for each population *i* (X_i)
for each population *i* do
Evaluate the fitness function (f_i)
end for

Table 5.2: PermGA pseudocode to generate an OLH design

For illustration, Figure 5.4 shows an OLH design of 100 points generated using Equation 5.1. To illustrate space filling properties of the OLH design, the *Euclidean* distance from the nearest point [80], [81] (as discussed in Chapter 2) was shown in Figure 5.5 for each of the test points. This figure shows that the

test points are distributed within the design space maintaining the space filling principle, since none of the generated test points are located too close to each other within the design space.

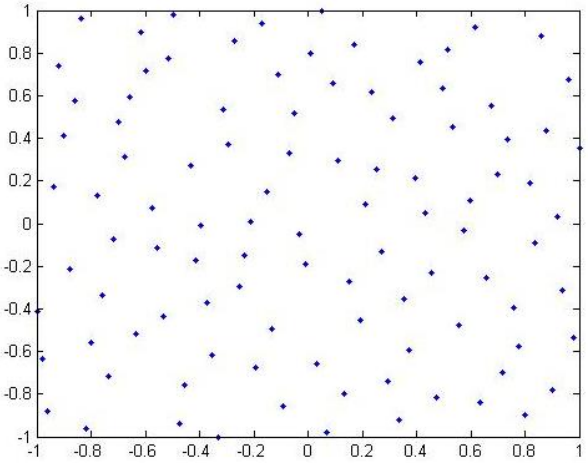


Figure 5.4: OLH of 100 points

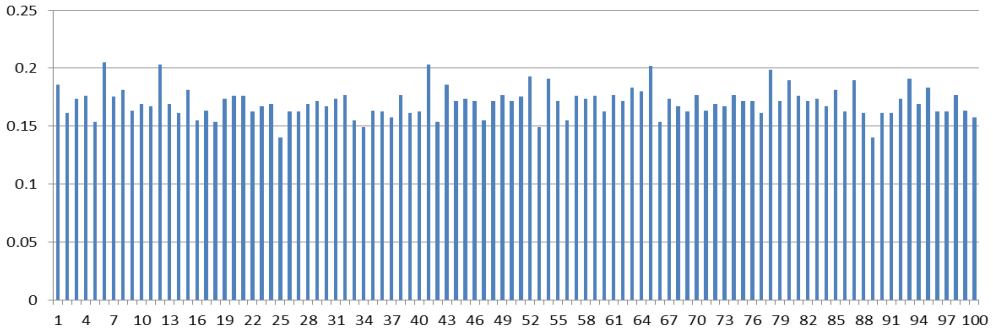


Figure 5.5: Euclidean minimum distance for 100 test points

The correlation among the 2 variables, i.e. as a measure of design orthogonality, was evaluated for this example. The yielded correlation value was (-0.003), which confirms the final design is quasi-orthogonal.

Figure 5.6 illustrates a typical convergence plot in terms of fitness for the PermGA algorithm. This algorithm was run for 1,000 iterations (i.e. set condition for algorithm termination) while the ‘Population Size’ was 200, ‘Crossover Rate’ was 0.8, ‘Mutation Rate’ was 0.05, and ‘Elite Size’ was 5.

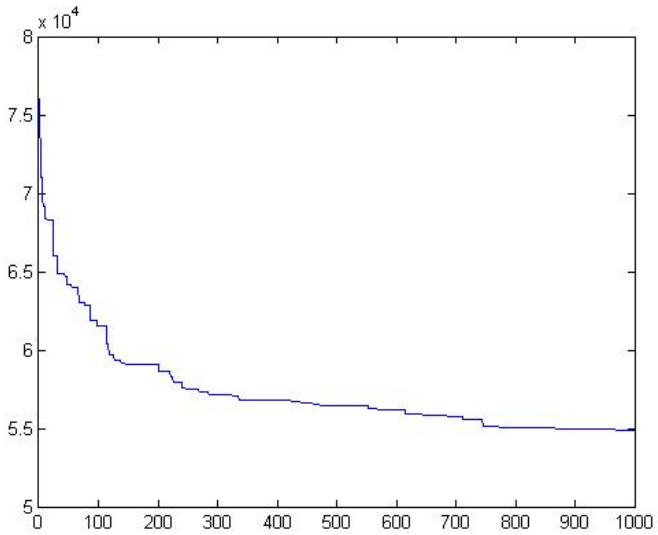


Figure 5.6: Convergence plot of PermGA

5.4.2 PermGA Modification for MB-MV DoE Augmentation

For the optimal augmentation of the DoE, i.e. by generating optimal 'infill' points for the MV set, the same uniformity metric can be used, the AELH function, by using the PermGA algorithm. The main challenge for this step is to formulate the objective function to take into account the position of the MB points already fixed in the design space. This means that the fitness function should include both the *P* new test points and the existing *M* test points from the MB OLH DoE. The fitness function was modified accordingly, as shown in Equation 5.2.

$$U = \left(\sum_{p=1}^{P}\sum_{q=p+1}^{P}\frac{1}{{L_{pq}}^2} + \sum_{p=1}^{P}\sum_{m=1}^{M}\frac{1}{{L_{pm}}^2}\right) \qquad \text{Equation 5.2}$$

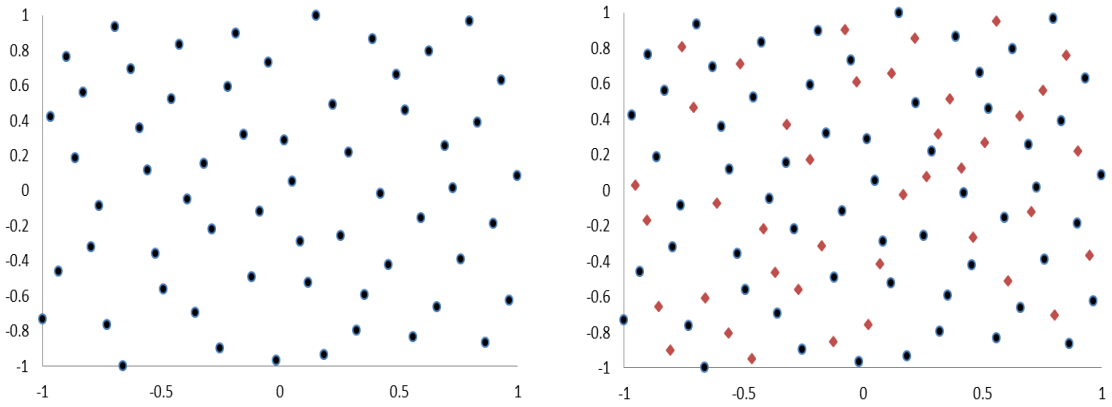
where *U* is the potential energy, L_{pq} is the distance between the points *p* and *q* where ($p \neq q$), and L_{pm} is the distance between each new point *p* and the exiting points *m*. The outcome of the MB-MV step design will be an OLH design with an optimal uniform distribution of points across the design space, with the new MV points optimally filling the under-sampled areas in the original design.

Figure 5.7 shows the result of an MB-MV DoE sequence for a 2 dimensional problem, with 60 DoE points (circle dots) generated using Equation 5.1, and 40 MV infill DoE test points (diamonds) generated using Equation 5.2. Figure 5.8 illustrates the space filling properties of the merged MB-MV DoE, in terms of the *Euclidean* distance for each of the test points. Accordingly, it can be argued that

none of the generated test points through 2 sequences, i.e. a MB design followed by a subsequent MV design, are located too close to each other within the design space.

The correlation among the variables, i.e. design orthogonality, was also studied for this example. The correlation between vectors of variables for 60 OLH DoE points is (-0.03), and for the total design of 100 test points, after adding the 40 MV DoE test points, is (-0.01), which means the final design is quasi-orthogonal.

Figure 5.9 illustrates a typical convergence plot in terms of fitness for the MB-MV algorithm. This algorithm was run for 1,000 iterations, while the 'Population Size' was 200, 'Crossover Rate' was 0.8, 'Mutation Rate' was 0.05, and 'Elite Size' was 4.



(a) (b)

Figure 5.7: MB-MV sequence: (a) MB, OLH of 60 points, (b) diamond points show the position of validation points (MV), OLH of 40 points, among the circle MB points.

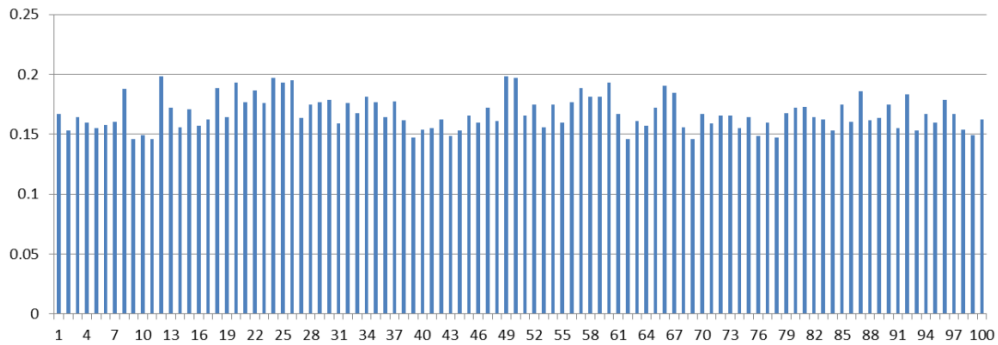


Figure 5.8: Euclidean minimum distance for all MB-MV test points (100 tests)

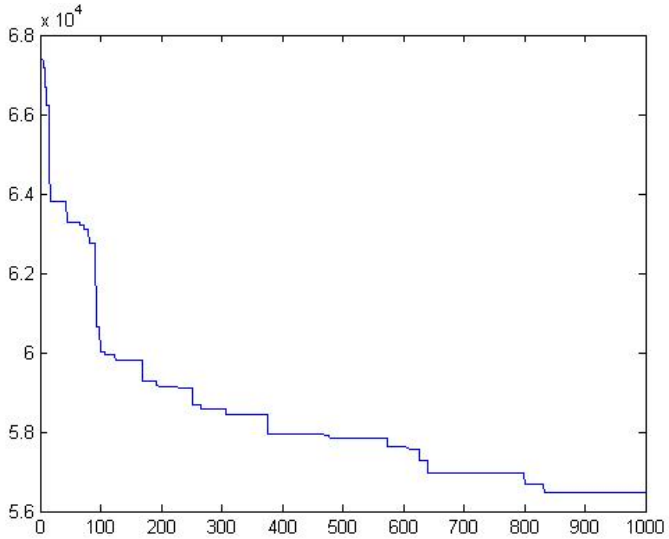


Figure 5.9: Convergence plot of PermGA for generating the MV OLH DoE

5.4.3 PermGA Modification for Constrained Variable Space

Equation 5.1 and Equation 5.2 are capable of generating the OLH test points within a symmetric design space in each of the dimensions. For many engineering problems the design space might be constrained in relation to some design variables (i.e. asymmetric design space). This could impair the ability of the PermGA algorithm to generate enough valid points for the MB, or any of the subsequent MVs, and affect the space filling quality of the generated design.

A variety of constraint-handling methods for evolutionary algorithm have been proposed in the last decades. Michalewicz and Schoenauer [234], and Mezura & Coello [235] have summarized most of the evolutionary constraint handling methods. The commonly used methods are:

1) *Repair* strategy, the idea of this strategy is that an infeasible individual is *repaired* to a feasible individual [236].
2) *Sudden Dead* strategy (also called *Dead penalty*), in which an infeasible individual is *removed* immediately from the population [237].
3) *Penalty functions*, the basic idea of this strategy is to define the fitness functions by extending the objective function with a penalty term. Penalty functions are the most commonly used approaches for evolutionary algorithm, in particular for handling inequality constraints [238].

In this thesis, a penalty term ∅ was added to the objective function in Equation 5.2, as shown in Equation 5.3, to avoid unnecessary computational costs by generating test points that are not feasible or do not have a physical meaning [94]. The form of the penalty term is shown in Equation 5.4. The penalty function is calculated based on the distance between the location of the test point O_p and the location of constraint limit O_l.

$$U = \left(\left[\sum_{p=1}^{P}\sum_{q=p+1}^{P}\frac{1}{L_{pq}{}^{2}} + \emptyset\right] + \left[\sum_{p=1}^{P}\sum_{m=1}^{M}\frac{1}{L_{pm}{}^{2}}\right]\right) \quad \text{Equation 5.3}$$

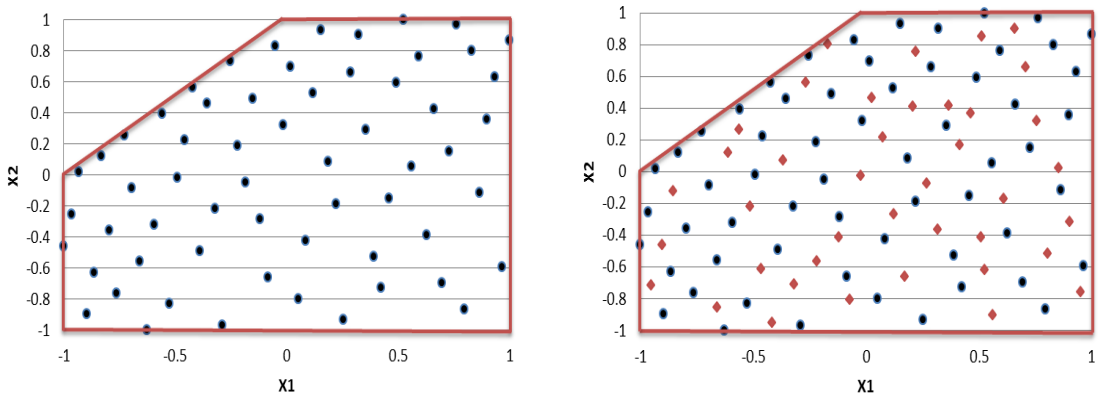
$$\emptyset = \begin{cases} 0 \ , \text{ if constraints are met} \\ e^{|O_p - O_l|}, \text{ if constraints are not met} \end{cases} \quad \text{Equation 5.4}$$

Figure 5.10 illustrates the results of an MB-MV DoE sequence for a linearly constrained two-dimensional problem, with 60 MB DoE points (illustrated by black circles), and 40 MV DoE test points (illustrated by red diamonds), both generated using the augmented infill PermGA algorithm, with the fitness function given in Equation 5.3. In this example, *X1* and *X2* can vary between −1 to 1, subject to the constraint ($X2 - X1 < 1$). This constraint has resulted in an asymmetric design space, as illustrated in Figure 5.10.

The space filling uniformity for the MB-MV OLH DoE points in the design space is illustrated in Figure 5.11, based on the *Euclidean* distance to the nearest test point. This figure shows that the infill points are distributed within the constrained design space preserving the space-filling principle. The correlation between vectors of design variables, or design orthogonality, was also investigated for this algorithm. It was seen that the correlation between vectors of *X1* and *X2* parameters for 60 OLH DoE points is (0.12), and for the total design of 100 test points is (0.15), which is still an acceptable value for correlation term, i.e. within the [-0.3 0.3] limit suggested by Steinberg and Lin [90].

Figure 5.12 illustrates the convergence plot in terms of fitness for the augmented MB-MV algorithm, showing that the augmented search algorithm is smoothly converged to the optimum solution. This algorithm was run for 1,000

iterations with ‘Population Size’ 200, ‘Crossover Rate’ 0.8, ‘Mutation Rate’ 0.05, and ‘Elite size’ 4.



(a) (b)

Figure 5.10: MB-MV sequence for asymmetric design space: (a) MB, OLH of 60 points, (b) diamond points show the position of validation points (MV), OLH of 40 points, among the circle MB points.

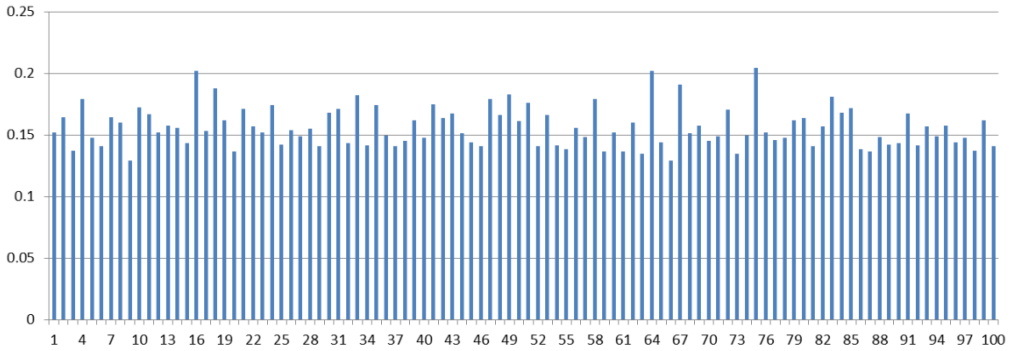


Figure 5.11: Euclidean minimum distance for all MB-MV test points (100 test points)

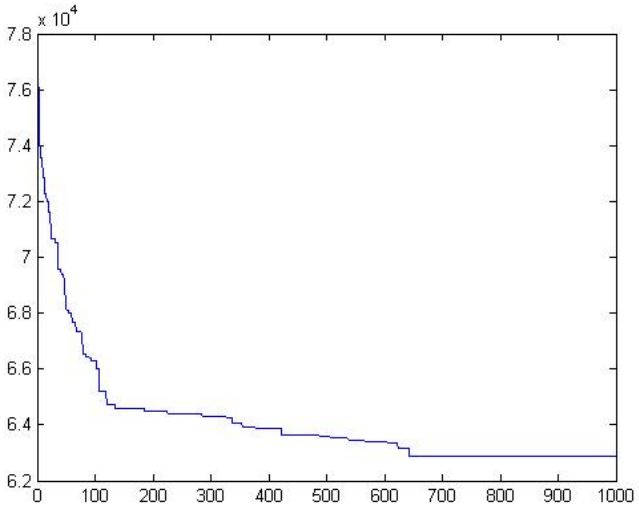


Figure 5.12: Convergence plot of PermGA for generating MV design of 40 points

5.5 Preliminary Validation of the MB-MV DoE Strategy

This section describes the application of the MB-MV DoE sequence to a surrogate engineering problem in order to validate the method and the algorithms developed prior to full scale implementation on an engine test experiment. The surrogate engineering problem is the well-known six-hump-camel-back (SHCB) mathematical function used for evaluating the global optimisation plans [229], given in Equation 5.5. This function has a complex shape with six local optima and two global optima of -1.0316 at (0.0898,-0.7127) and (-0.0898, 0.7127).

$$f(x) = 4{x_1}^2 - 2.1{x_1}^4 + {}^1\!/_3\,{x_1}^6 + x_1x_2 - 4{x_2}^2 + 4{x_2}^4$$

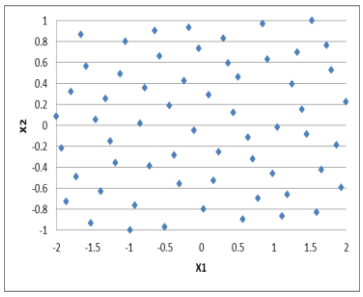
where $x_1 \in [-2,2]$, $x_2 \in [-1,1]$ Equation 5.5

The MB-MV DoE strategy was applied in four iterations. In the first step, an MB OLH DoE with 60 points was generated using developed PermGA algorithm. A MV OLH DoE with 15 points was generated as the first Model Validation design (MV1). The SHCB function was evaluated at both MB and MV1 points, and a response model was fitted based on the MB test points in the MATLAB MBC Toolbox™, using Radial Basis Functions (RBF) models [239] (i.e. RBF with Thinplate Kernel function [121]). Both internal validation criteria (based on PRESS RMSE [224]) and external model validation criteria (based on RMSE [93] for the MV1 validation set, as given by Equation 2.15) were considered. In step two, the same type of RBF response model was built based on the joint MB+MV1 test points (i.e. 75 points). A second Model Validation (MV2) was generated based on a 15 points OLH MV DoE. The same process of internal and external validation (MV2 points) was applied. This process was repeated with two further iterations, with MV3 = 15 points and MV4 = 15 points (i.e. in the 4th iteration the model building set comprised MB+MV1+MV2+MV3 = 105 points, an MV4 = 15 points). The results from this experiment are summarised as follows:

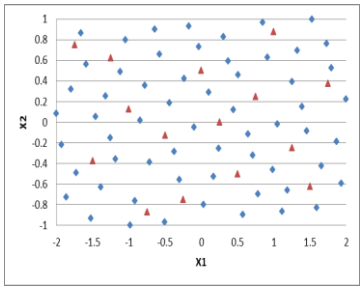
Figure 5.13 illustrates the distribution of points in the DoEs at each iteration. Figure 5.14 shows the uniformity of the distribution of the points in the design space, in terms of the *Euclidian* distance for each of the 120 test points in the

joint MB-MV DoEs. This histogram indicates that distribution of the Euclidian distance for the test points is quasi-uniform and even after 4 independent steps of testing the distributed points are still remote from each other within the design space (i.e. test points are not replicated). For better illustration of the uniformity of the distributed test points within the design space (i.e. space filling properties), the Euclidian distance of the test points across the subsequent MV DoEs are illustrated using boxplots, as shown in Figure 5.15. This figure indicates that:

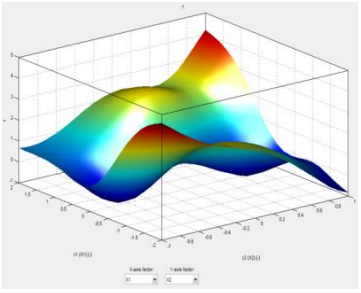
1) The *Euclidian* distance of test points decreases by adding more test points (from MB-MV1 to MB-MV4), which was expected since the number of test points within the finite design space are increasing over the MV DoE sequences.
2) The variability of the *Euclidian* distances (i.e. the interquartile range [240]) decreases across the subsequent DoEs. This trend demonstrates the ability of PermGA algorithm to enhance the space filling properties of the merged DoEs, by preserving the uniform distribution of test points when generating the MV DoEs.
3) A number of outliers observed by collecting more data (i.e. 'MB-MV3' and 'MB-MV4'). An outlier indicates that the Euclidian distance of the test point is beyond the upper and lower whiskers. The whiskers are defined as the observations that are at least 1.5 times the interquartile range from the edges of the box (i.e. which are showing quartile 1 and 3) [240]. The main reason for appearance of the outliers is that the spread of the Euclidian distances is smaller for the 'MB-MV3' and 'MB-MV4' DoEs, due to a more uniform distribution of Euclidian distances. Accordingly, the interquartile range, and consequently the whiskers, is smaller in these DoEs, which in effect increases the possibility of having more test points with Euclidian distances out of the whiskers' range (i.e. outliers).



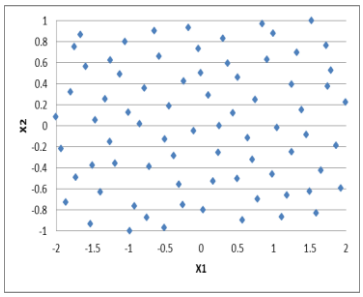
Stage 1: B=60 points



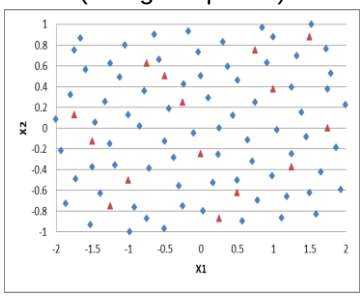
MV1=15 points
(triangular points)



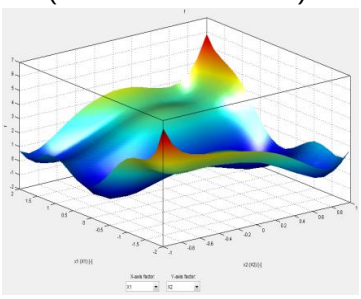
RBF Model
(PRESS RMSE: 0.43)



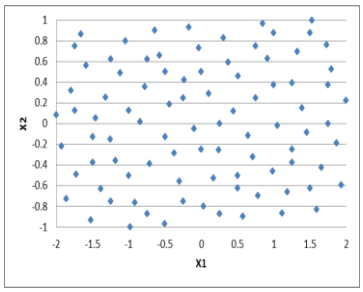
Stage 2: MB=75 points



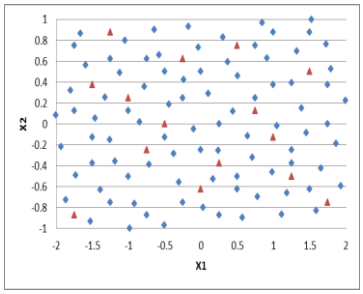
MV2=15 points
(triangular points)



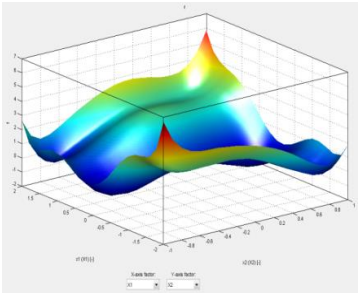
RBF Model
(PRESS RMSE: 0.1)



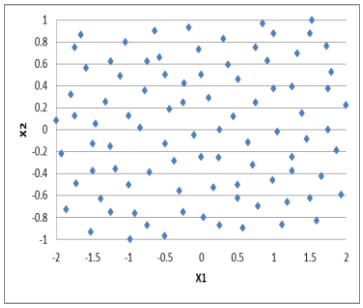
Stage 3: MB=90 points



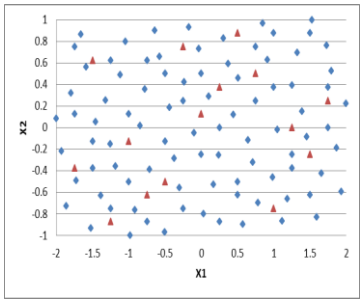
MV3=15 points
(triangular points)



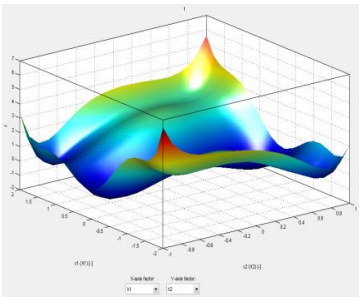
RBF Model
(PRESS RMSE: 0.05)



Stage 4: MB=105 points



MV4=15 points
(triangular points)



RBF Model
(PRESS RMSE: 0.03)

Figure 5.13: MB-MV DoE projection and response surface modelling for SHCB problem

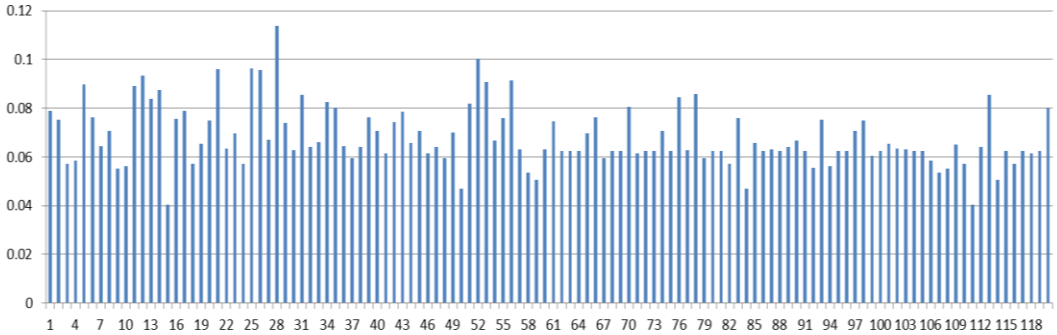


Figure 5.14: *Euclidean* 'Maxi-min' (Mm) distances of all the sample points in four steps

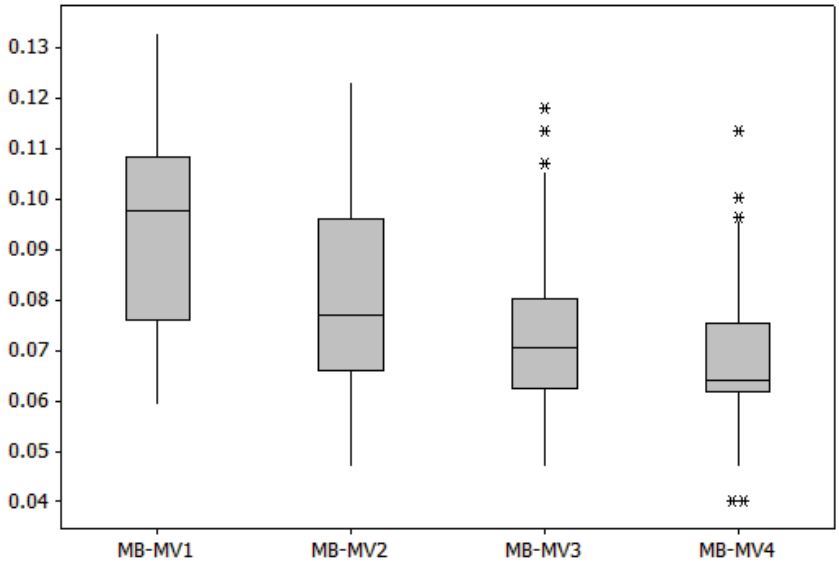


Figure 5.15: Boxplot of *Euclidean* distances across the subsequent validation DoEs

Figure 5.16 characterises the distribution of the Euclidean distances in terms of its standard deviation across the 4 stages. This graph shows that the uniform space filling properties of the MB-MV DoEs are improving over the subsequent stages of DoE. The correlation between *X1* and *X2* after 4 stages was also calculated using Equation 2.5 as $r = 0.05$, which indicates that the final design is quasi-orthogonal.

Furthermore, Figure 5.13 gives a graphical illustration of the response surfaces fitted at each stage. These graphs clearly show that the accuracy of the model improves through the successive MB-MV stages. By looking at the internal model validation criterion (PRESS RMSE, given in Figure 5.13) and external validation (RMSE for the MV test points, shown in Figure 5.17), it can be

concluded that the accuracy of the model improves dramatically over the first 3 stages, with only a small improvement between the 3rd and 4th stage.

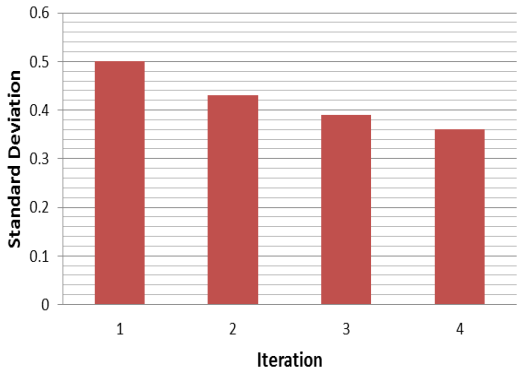


Figure 5.16: Standard deviation (σ) of the Mm Distance (Stages 1 to 4)

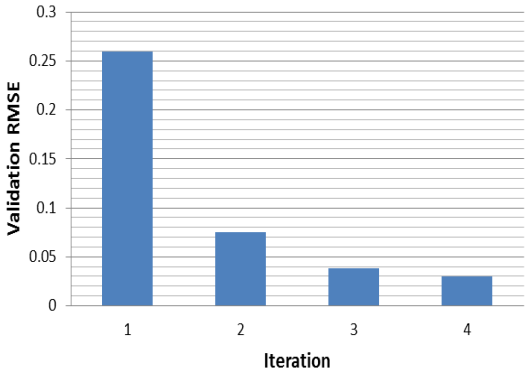
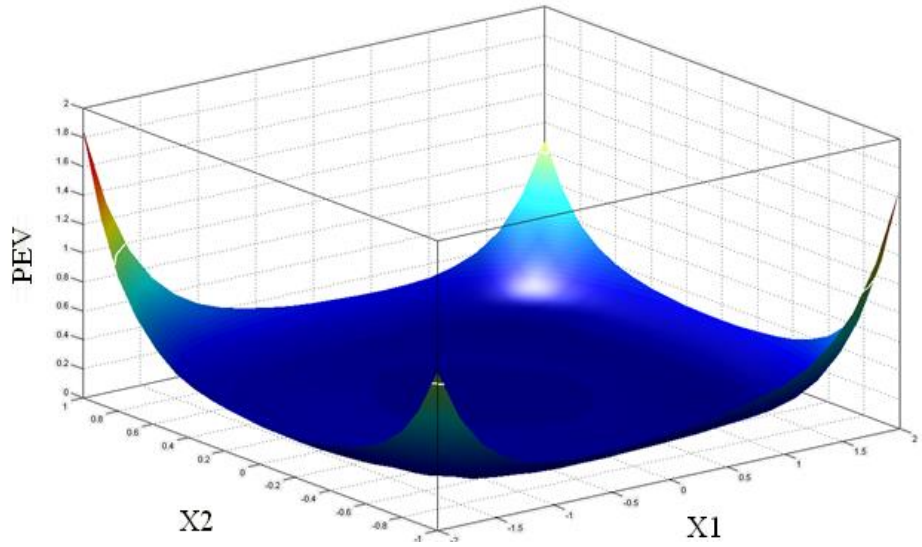
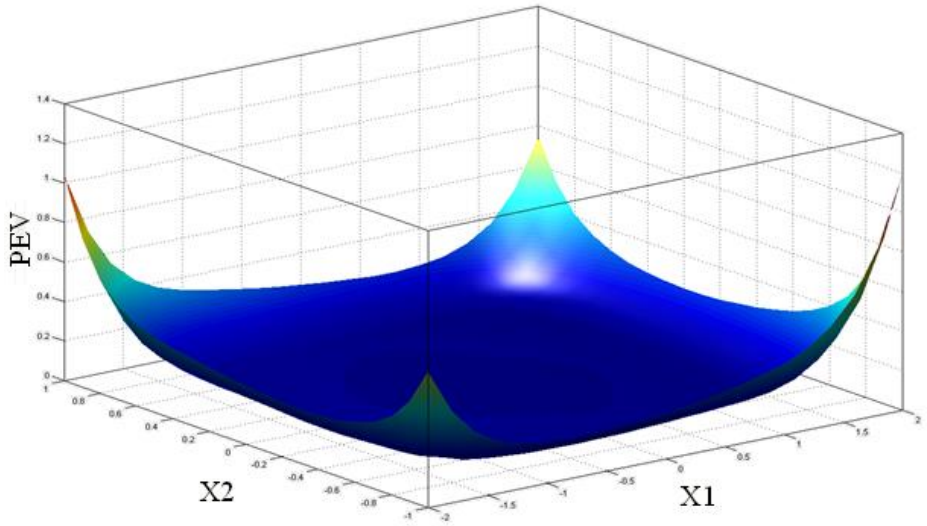


Figure 5.17: External validation of the built models through MB-MV sequence (Stages 1 to 4)

Generally, the Predictive Error Variance (PEV) [71] for LH designs is high around the boundary limits [156] which means model accuracy is poor in that area, i.e. particularly for non-parametric modelling techniques (such as RBFs) which would face difficulty to extrapolate outside the area where test points are collected [49]. The initial design in MB-MV strategy is also a LH design. However, due to employing LH DoE technique to generate the successive validation sample batches, which distributes the points within design space using gridding the space to equal intervals, more points will be collected iteratively from the grids next to the boundary limits. This data collection strategy results in collecting more data from the areas next to boundary limits with lower model quality, which consequently enhances the model quality in these areas. The PEV of the experimental plans for the SHCB function at stages 1 and 4 are illustrated in Figure 5.18. It can be seen that the PEV next to the boundary limits are high at stage 1 (i.e. around 1.8), which is reduced by 40 % while adding up the subsequent validation designs.



a) PEV for SHCB function at stage 1, using a RBF model



b) PEV for SHCB function at stage 4, using a RBF model

Figure 5.18: PEV graph for SHCB function

The main conclusion from this study was that the proposed MB-MV sequential DoE framework is successful at generating a quasi-orthogonal DoE with good space filling properties. The proposed design is also a fine-grained design, augmented iteratively with small batches of MV points, with good projection properties, since it uses batches of OLH designs to cover the whole range of design space for each parameter.

Therefore, the MB-MV DoE framework should be applicable for engine mapping experiments, as it can support the development of good engine response models based on space filling mapping DoEs. It also has the potential to improve the test efficiency, because the testing can be stopped when the

desired model accuracy is reached, instead of the current fixed size DoE plans which do not guarantee that a satisfactory model is achieved.

5.6 Implementation of the MB-MV DoE Strategy for the GDI Engine Case Study

This section describes implementation of the proposed MB-MV DoE strategy for the AJ133 GDI engine mapping case study, illustrated in Figure 4.4. The engine mapping research for the GDI engine case study was conducted in two phases:

1- Single injection phase (base-line strategy): for all 6 steady state minimap operating conditions shown in Table 4.4.

2- Multiple Injections phase (double-injection strategy): for 4 of the steady state minimap operating conditions, i.e. minimap points 3 to 6 as shown in Table 4.4, for which double injections is accessible.

For each of these two phases, the same DoE strategy was employed with the aim of delivering engine response models of sufficient accuracy for the calibration optimisation stage.

5.6.1 Engine Mapping Methodology

Given the relatively low state of knowledge about the effect of GDI engine calibration variables on Pn, the strategy adopted in this thesis was to pursue a wider exploration of the variables space in search for potential areas of trade-off between the key attributes (i.e. engine responses of interest). This pointed to a two-stage DoE process: screening stage and engine mapping stage.

5.6.1.1 Screening Stage

The main aim of the screening DoE was to:

- Evaluate the effect of calibration variables on the key engine responses (Pn and fuel consumption) in terms of significance, trends and interactions.
- Define the valid design space for the calibration variables at each of the steady state engine speed / load operating points.

The common strategy for the screening stage is to use test automation software (such as AVL CAMEO^{TM}) to quickly identify the design space for each

parameter through carrying out centre-out exploration experiments, i.e. via a self-adapting online DoE to define the limits of calibration parameters [241], [242]. However, these automated testing strategies have the limitation that the information generated through the exploration tests is not further utilised for the model building experiments.

The strategy adopted in this research was to use a space filling OLH DoE of a smaller size (e.g. 40-60 test points), in order to support an effective design space exploration for potential areas of calibration trade-off [65]. A good space filling DoE strategy can be a good candidate for the screening stage since the uncertainty quantification of the predicted value at a test point is influenced by the distance between the point and the other generated test points within the design space [96]. This OLH design was generated using the developed PermGA algorithm discussed in Section 5.4, based on Equation 5.1.

The collected DoE test data was processed in three steps:

I. **Combustion stability analysis:** evaluation of the GDI engine operating constraints at each minimap point, as discussed in Chapter 4, to identify the unstable test points;

   a. *Combustion stability constraint*: where the combustion stability exceeds the combustion stability set threshold as below;

      - CoV(IMEP) less than 3%, for NMEP > 3bar,
      - sdNMEP less than 0.1 bar, for NMEP < 3bar).

   b. *Exhaust gas temperature* (T_{ext}): where the exhaust temperature is more than 800 ^{o}C.

   The unstable test points should be excluded from the further analysis (or model fitting stage), since it is expected that the infeasible points with unstable combustion conditions to have significantly higher variability than ‘stable’ points. This feasibility evaluation was used to define the final boundary limits of the calibration parameters for planning of the mapping stage DoE.

II. **Engine response surface models:** fit response surface models for the valid test data using the MATLAB MBC Model Browser toolbox. The fitted models to the screening test points are low fidelity, i.e. not sufficiently accurate for prediction purposes. However, they can be used to explore and validate the engine response trends, study the interactions between calibration variables, and identify the valid range for the actuators in the multidimensional design space.

III. **Trend Analysis:** study and validate the engine response trends both statistically and through engineering judgment. This consisted of both analysis of models and trends at a given engine speed / load point, as well as consistency of trends across the test engine speed / load space.

5.6.1.2 Engine Mapping Stage

Once the screening stage has been completed, the mapping stage begins with the aim of planning and running detailed DoEs to fit accurate engine response surface models in the variable space defined by the screening stage, to be used for the calibration optimisation study. To fulfil the model accuracy requirements an efficient DoE strategy is essential to ensure that the target model accuracy is obtained with minimum testing effort. So, the MB-MV framework was implemented to generate sequential test plans as OLH DoEs generated using the PermGA algorithm. The main model building DoE was based on a 50 test points OLH DoE, with further 15 test points collected for the model validation generated with the MB-MV algorithm, Equation 5.2 or Equation 5.3 (based on the constraints from the screening stage). The test plan was then iteratively augmented by subsequent MV's of size 15, until the model accuracy requirement was achieved for all the engine responses. The MATLAB MBC Model Browser toolbox was used to fit the response models at each stage. The best response model, which could truly represent the corresponding GDI engine response, was then selected from a series of candidates (i.e. Polynomials and RBFs with different Kernel functions) based on:

I. **Analysis of residuals**: analysis of patterns of residuals (i.e. the difference between the predicted values by the response surface model and the

collected data) for the MB set, and of the prediction errors for each MV set.

II. **Statistical diagnostics:**

   a. *External model validation*: using Validation RMSE, i.e. RMSE [156] for prediction errors of the new test data in the validation set (see Equation 2.15), and Relative Error (see Equation 2.16).

   b. *Internal model validation*: using PRESS RMSE [224] for the MB set (i.e. root mean square of the prediction sum of squared errors for the MB set based on simple cross-validation).

III. **Trend analysis:** Engineering analysis of engine response trends through investigating of the GDI engine physical behaviour, to ensure that the models are not over-fitted or extrapolating among the collected data.

An important feature of the proposed engine mapping methodology is that the screening stage DoE was taken as the first step in the engine mapping DoE sequence. So, the feasible test points that were inside the revised variable space after the screening stage were carried over to the mapping stage to enhance the strategy efficiency. The overall engine mapping procedure for the GDI engine is illustrated in Figure 5.19.

The following sections present the screening and mapping DoE stages, which were conducted for the GDI AJ133 engine case study using single-injection and double-injection strategies.

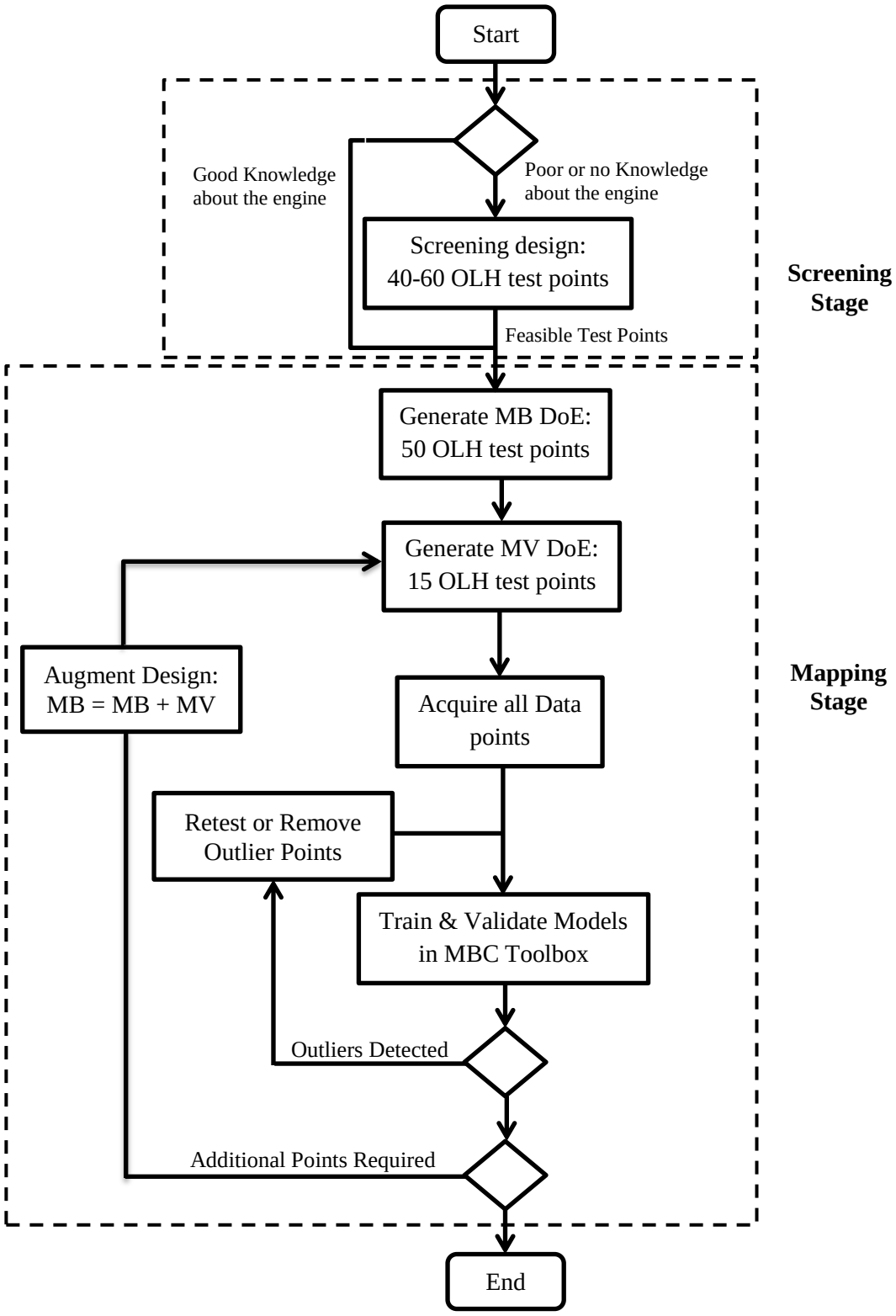


Figure 5.19: The offline DoE and modelling strategy proposed for the GDI engine

5.6.2 Single-Injection Phase

5.6.2.1 Screening Stage DoE: Implementation

The screening stage for the AJ133 GDI engine with single injection strategy was conducted through generating an OLH design of 40 test points over the 4 calibration variables (IVO, EVC, FRP and SOI). The design range for each calibration parameter is shown in Table 5.3.

At each of the 40 test points, a 3 point spark sweep was carried out as follows:

1) 'Nominal': MFB50 (50% Mass Fraction Burned) position at 8°ATDC.
2) 'Early/advanced': MFB50 + 4^{o}.
3) 'Late/retarded': MFB50 - 4^{o}.

Spark timing was included in the experiment to evaluate whether early / late spark could have a beneficial effect on particulate emissions (Pn). The nominal spark setting is taken to be '50% burn at 8° ATDC' (MFB50) which was possible to set using the instrumentation available on engine, i.e. in-cylinder pressure sensors fitted to all cylinders.

Code	**Name**	**Range**		**Unit**
FRP	Fuel Rail Pressure	3	15	bar
SOI	Start of Injection	220	335	deg BTDC
IVO	Inlet Valve Opening	-12	50	deg ATDC
EVC	Exhaust Valve Closing	-6	44	deg ATDC
ST	*Spark Timing*	*MFB50 - 4*	*MFB50 + 4*	*deg ATDC*

Table 5.3: Engine control variables for the GDI engine with single-injection strategy

The distribution of Screening DoE test points within the 4-dimensional design space is illustrated in Figure 5.20. This DoE was generated by running the developed PermGA, based on Equation 5.1, for 1,000 iterations, when the 'Population Size' was 200, 'Crossover Rate' was 0.8, 'Mutation Rate' was 0.05, and 'Elite Size' was 4. The PermGA convergence plot is shown in Figure 5.21. Figure 5.22 shows the Euclidean minimum distance for the 40 Screening test points, i.e. with normalised parameters between [0 1], to illustrate the space filing properties of the generated OLH design within the 4-dimensional design

space. Accordingly, it can be argued that the test points are distributed quasi-uniformly and none of the points are located too close to each other within the design space. Moreover, the correlation between each of the two variables was studied for the Screening design. The maximum correlation value was (0.04), between IVO and FRP, which essentially means that the generated OLH design is quasi-orthogonal.

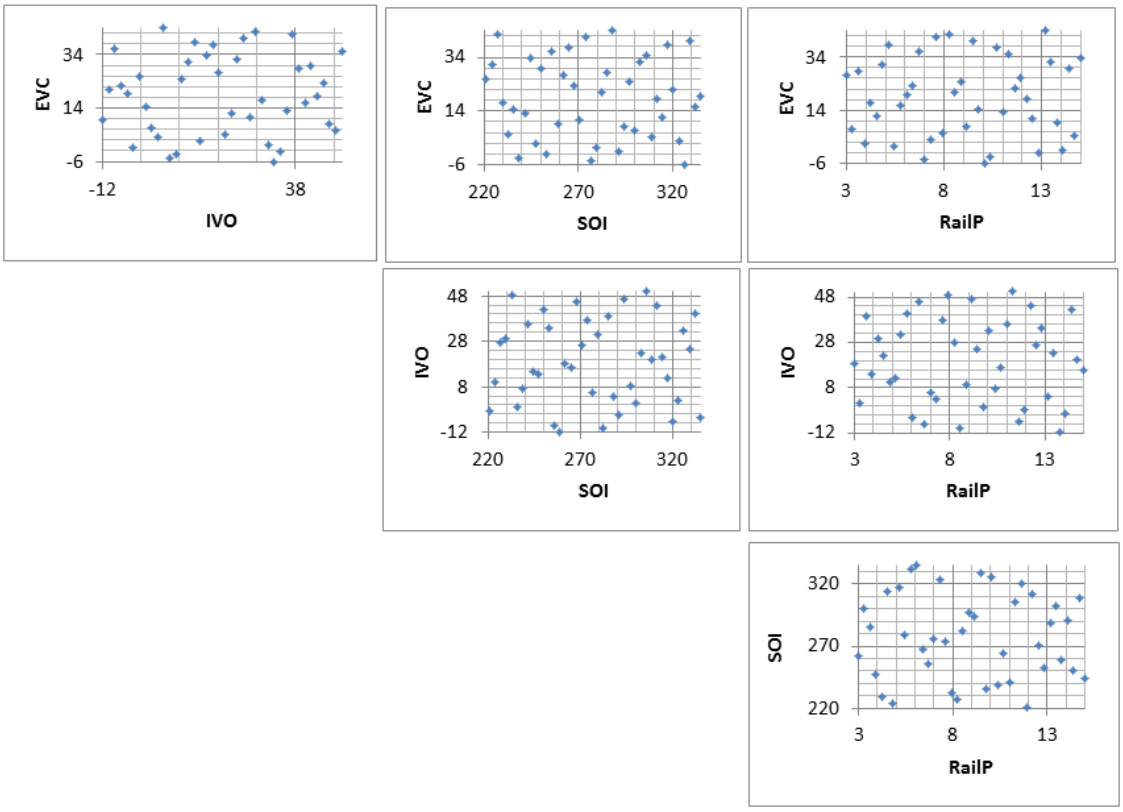


Figure 5.20: Distribution of 40 Screening DoE test points in design space

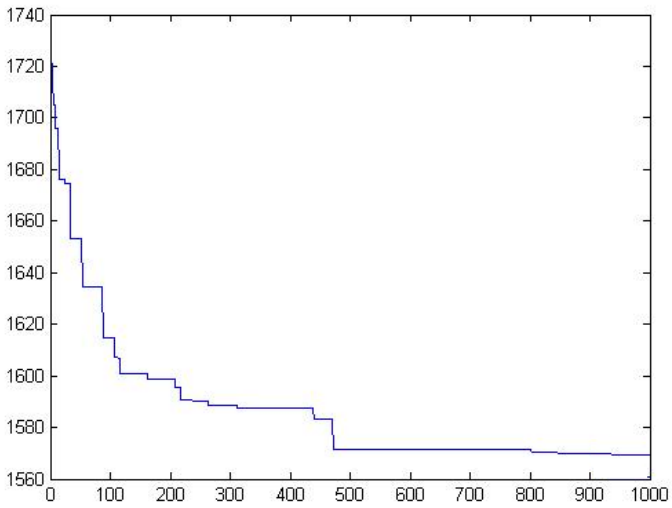


Figure 5.21: PermGA Convergence plot for generating the Screening DoE

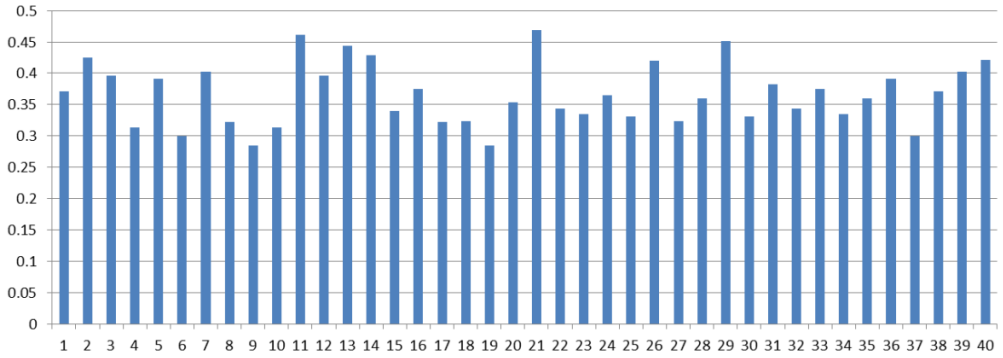


Figure 5.22: Euclidean minimum distance for the Screening DoE test points

5.6.2.2 Screening Stage DoE: Analysis of Results

The engine responses of interest, i.e. listed in Table 4.3, were collected for the Screening DoE test points. The analysis of the collected data was then carried out as following:

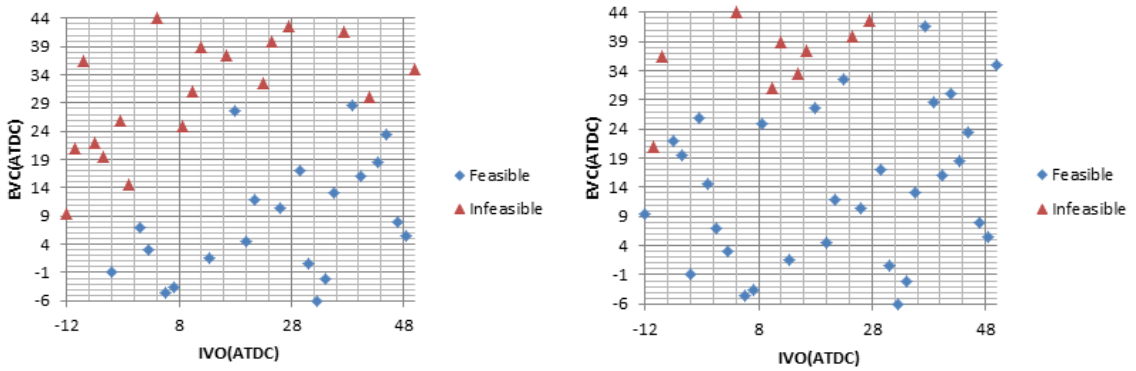
I. <u>Combustion Stability Analysis:</u>

A feasibility study was carried out on the test data to identify the points that do not meet the combustion stability set threshold, as summarised in Table 5.4. It was observed that the number of unstable points was less than 15% of the total test points for most minimap points tested, except the lower speed / load minimap points, i.e. '700 (rpm) / 28 (Nm)' and '1500 (rpm) / 41 (Nm)', where up to 45% of the DoE points were lost due to unstable combustion. The 'worst case' was the close to idle minimap point, '700 (rpm) / 28 (Nm)', where only 21 DoE test points were valid, i.e. at the nominal spark setting, from the combustion stability point of view. Figure 5.23, illustrates the distribution of infeasible points for the most problematic engine operating points, i.e. '700 (rpm) / 28 (Nm)' and '1500 (rpm) / 41 (Nm)', in 'EVC-IVO' coordinates. As expected, this shows a clear pattern, which can be explained based on the fact that late exhaust valve closing has a detrimental effect on combustion stability, particularly when IVO is early (illustrated by the infeasible top left corner in the plot). This is consistent with engineering judgment where at part load it is beneficial to retain some of exhaust gases in the cylinder by moving EVC timing after TDC, since this will cause a rise in internal EGR and also reduction in some exhaust emissions. However, there is a limit on how much EGR can be tolerated by the engine because it may have negative effects on combustion

stability, especially at lower loads where the charge density is low [218]. This can be shown more clearly by showing the distribution of the infeasible points in terms of the 'Overlap', i.e. $(EVC - IVO)$ versus EVC plot, illustrated in Figure 5.24. High overlap results in excessive EGR which has negative effects on combustion stability [217].

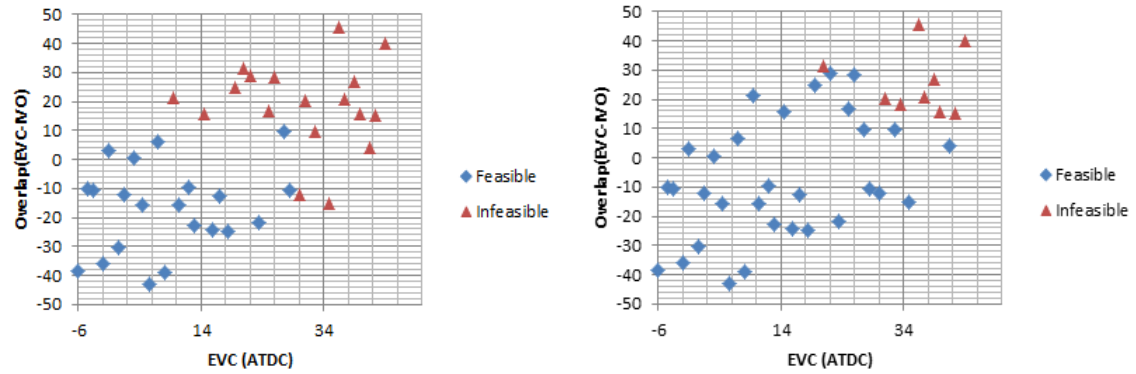
Test Point		ST Nominal MFB50	ST Late MFB50+4	ST Early MFB50-4
1	700 rpm / 28 Nm [sdNMEP]	21	20	24
2	1500 rpm / 41 Nm [sdNMEP]	31	28	37
3	1250 rpm / 125 Nm [CoV-IMEP]	36	36	36
4	1500 rpm / 105 Nm [CoV-IMEP]	36	36	36
5	2000 rpm / 81 Nm [CoV-IMEP]	35	35	35
6	2000rpm / 199 Nm [CoV-IMEP]	37	37	37

Table 5.4: Summary of valid points per minimap point tested



(a) (b)

Figure 5.23: Combustion stability for test data: (a) 700/28, (b) 1500/41 minimap points



(a) (b)

Figure 5.24: Combustion stability based on overlap: (a) 700/28, (b) 1500/41 minimap points

Based on this analysis, plots of overlap against EVC for all the minimap points (e.g. Figure 5.24) showed that the design space for IVO and EVC calibration variables could be reduced in the next step of DoE, i.e. particularly for the lower speed / load operating points. Therefore, the design space for IVO and EVC at '700 (rpm) / 28 (Nm)' minimap point constrained to EVC < 30 deg ATDC, and 'Overlap' < 10 deg. Also, at '1500 (rpm) / 41 (Nm)' engine operating point, the design space constrained to 'Overlap' < 30 deg.

II. Engine Response Surface Models:

Response surface models were fitted for each response at each operating point by using MATLAB MBC toolbox™, using only the valid test results summarised in Table 5.4. Noteworthy, separate models were fitted for each spark timing setting (nominal / early / late) at each speed / load test point. Figure 5.25 illustrates the MBC test plan / model setup.

The response model fitting procedure included the following steps:

1. *Fit candidate models*: several types of models (e.g. polynomial, RBF and Hybrid RBF) were fitted for each engine response. Given that a limited number of sample points were tested for the screening stage and the experimental range of calibration variables was deliberately chosen to be quite wide, polynomial models were unlikely to be efficient [96]. Therefore, a wide range of RBF models were considered to fit the response models. This flexile non-parametric modelling technique is expected to be more accurate at the points where they are interpolating due to its ability to confront the missing and noisy data with a good generalization capability [123]. The quality of the fit was preliminarily evaluated via RMSE [156] and PRESS RMSE [224], where necessary; extreme observations outliers have been eliminated from the model fitting process.

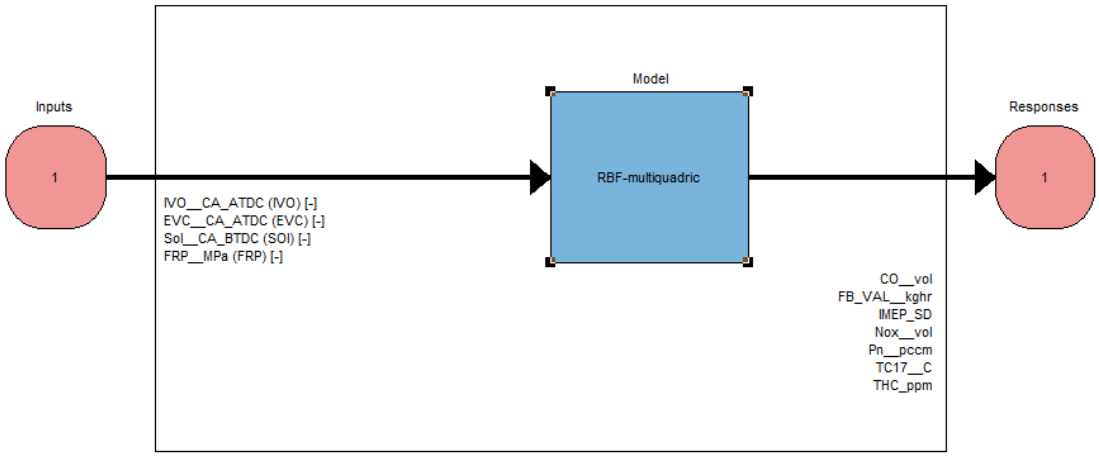


Figure 5.25: MBC model set-up

2. *Model selection*: given the small number of test points available for this Screening DoE, preference must necessarily be given to the fits that have a small modelling sequence (i.e. a small number of effective model terms). Therefore, in line with findings from established research [133], the Akaike Information Criteria (AIC) was used as the main model selection criteria. For engineering communication purposes, PRESS RMSE was still used to indicate roughly the prediction accuracy of the model.

   On review of the selected models, it was seen that RBF-multiquadric models showed better statistical properties than the other candidate models, therefore this was selected as preferred model type and fitted for all responses. Figure 5.26 and Figure 5.27 illustrate an example of results as a 3-D plot from MBC toolbox for fuel consumption (FC) and Pn responses at the 700 – 28 speed / load operating point, at nominal spark timing (MFB50% at 8^{o}). Similar models were fit for all the engine responses at all 6 engine steady state operating points, at 3 different spark timings (i.e. nominal / early / late spark timings).

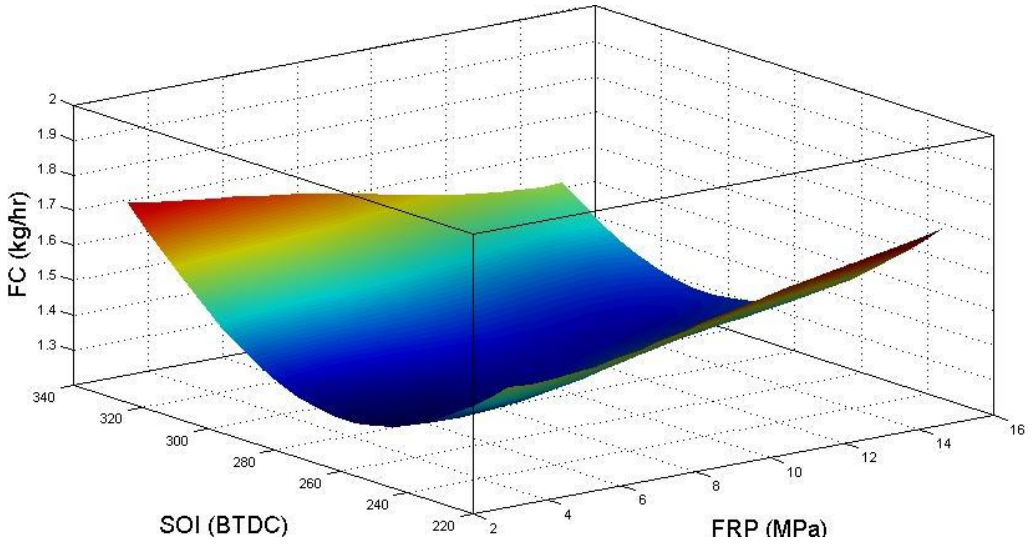


Figure 5.26: 3D View of Fuel consumption versus SOI and FRP for 700 / 28 minimap point

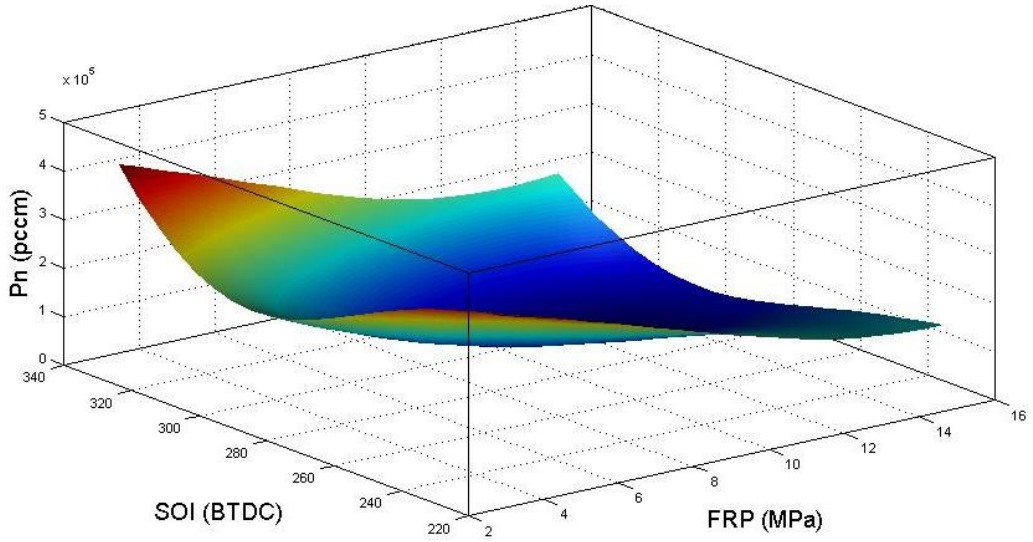
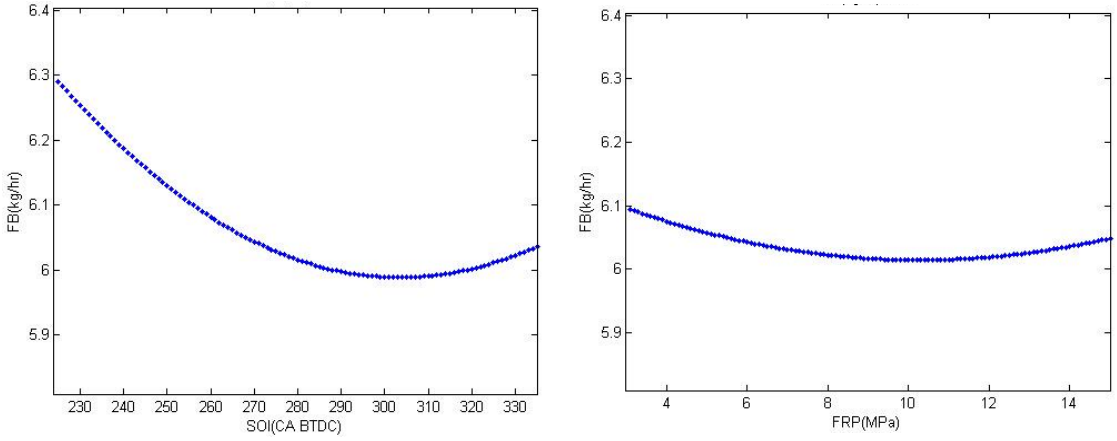


Figure 5.27: 3D View of Pn versus SOI and FRP for 700 / 28 minimap point

III. Trend Analysis:

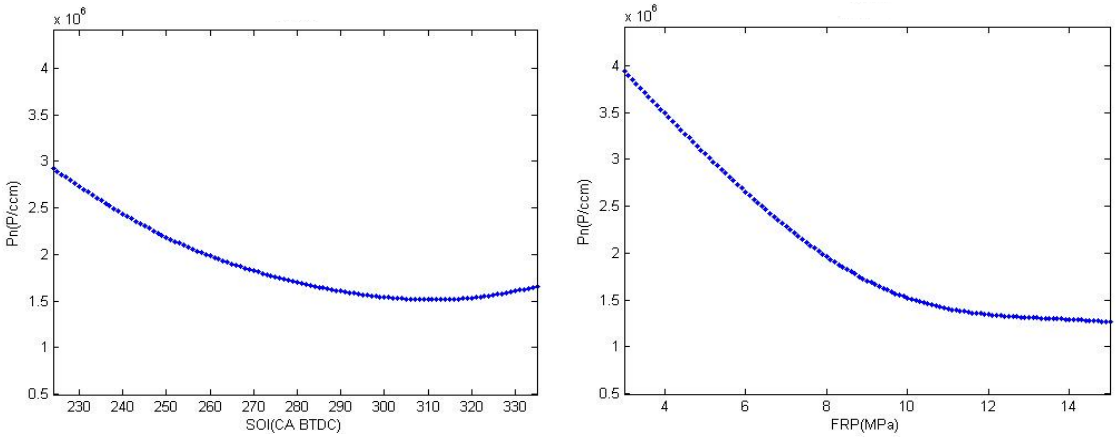
Figure 5.28 and Figure 5.29 illustrate examples of pseudo 'main effects' type plots, generated using Matlab programming software, based on RBF-multiquadric models. In these plots, fuel consumption and Pn responses are presented as a function of one factor-at-a time while the other factors are evaluated at their neutral position (i.e. the mid-range section of the response surface). Such plots are a useful method for the engineering evaluation of trends, to be used beside the 3D response models. By comparing the response plots at different operating points, it was seen that fuel consumption and Pn

responses are improving by increasing the FRP variable. This indicated that the FRP variable space could be reduced for the engine mapping stage. Similar analysis was done for the SOI calibration variable, it was seen that the higher range of start of injection variable, i.e. corresponding to 'early' SOI, is beneficial for both fuel consumption and Pn. So, the SOI variable space for engine mapping stage was also reduced for the engine mapping stage. Therefore, given that similar trend was observed for all the minimap points tested for the GDI engine, the final design space for FRP and SOI parameters were revised for the engine mapping stage to a subspace of original design space used for screening DoE, as illustrated by the rectangle in Figure 5.30.



(a) (b)

Figure 5.28: Fuel consumption against (a) SOI(BTDC), and (b) FRP(MPa), for the 2000-81 speed/load operating point at nominal spark timing



(a) (b)

Figure 5.29: Pn against (a) SOI(BTDC), and (b) FRP(MPa), for the 2000-81 speed/load operating point at nominal spark timing

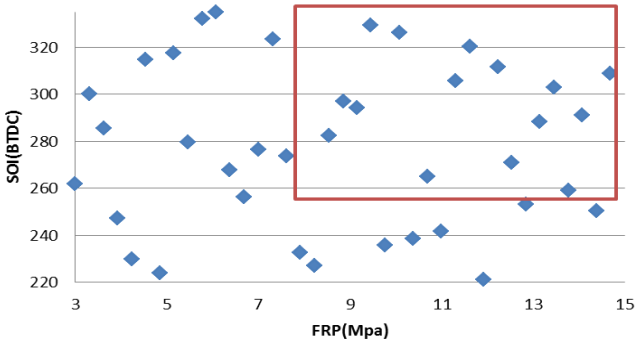


Figure 5.30: Design space for engine mapping stage

Similarly, Figure 5.31 illustrates an example of pseudo 'main effects' plots, in Matlab environment, for Pn with different spark timing strategies (Nominal, advanced and retarded). It was observed that although advancing spark timing shows a marginal Pn benefits over MBF50% burn spark timing, especially at '700 (rpm) / 28 (Nm)' operating point, the fuel consumption is more than MBF50% spark timing, as illustrated in Figure 5.32. Therefore, since no significant benefit was seen for any of the 6 engine speed / load operating points in performing the DoE at different spark timings, it was decided to fix spark timing at MFB50%.

The data analysis (i.e. combustion stability analysis, fitting response models, and trend analysis) was executed on the screening DoE data collected at all the 6 minimap points tested for the GDI engine with single injection strategy. All the results were collated in a technical report 'CREO-12032012: Preliminary Report on Phase 1 Screening DoE Results' [243].

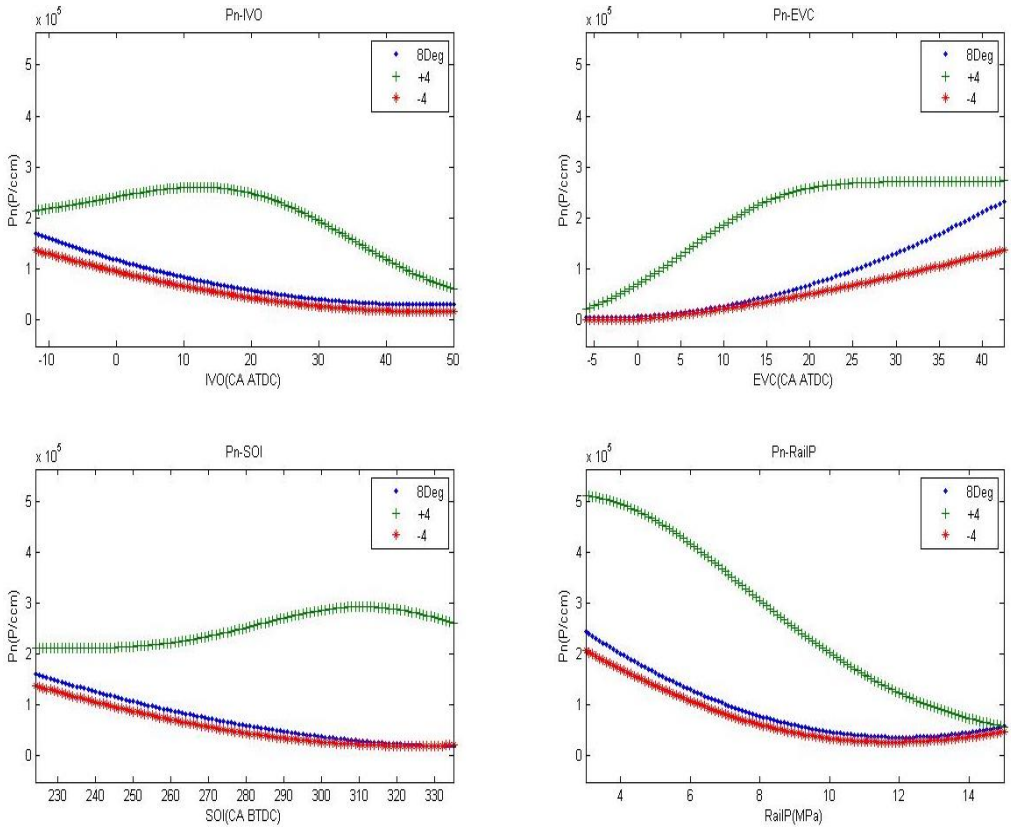


Figure 5.31: Effects of actuator settings on Pn at 700 – 28 operating point, Nominal is showed by a blue line, retarded by red and advanced by green.

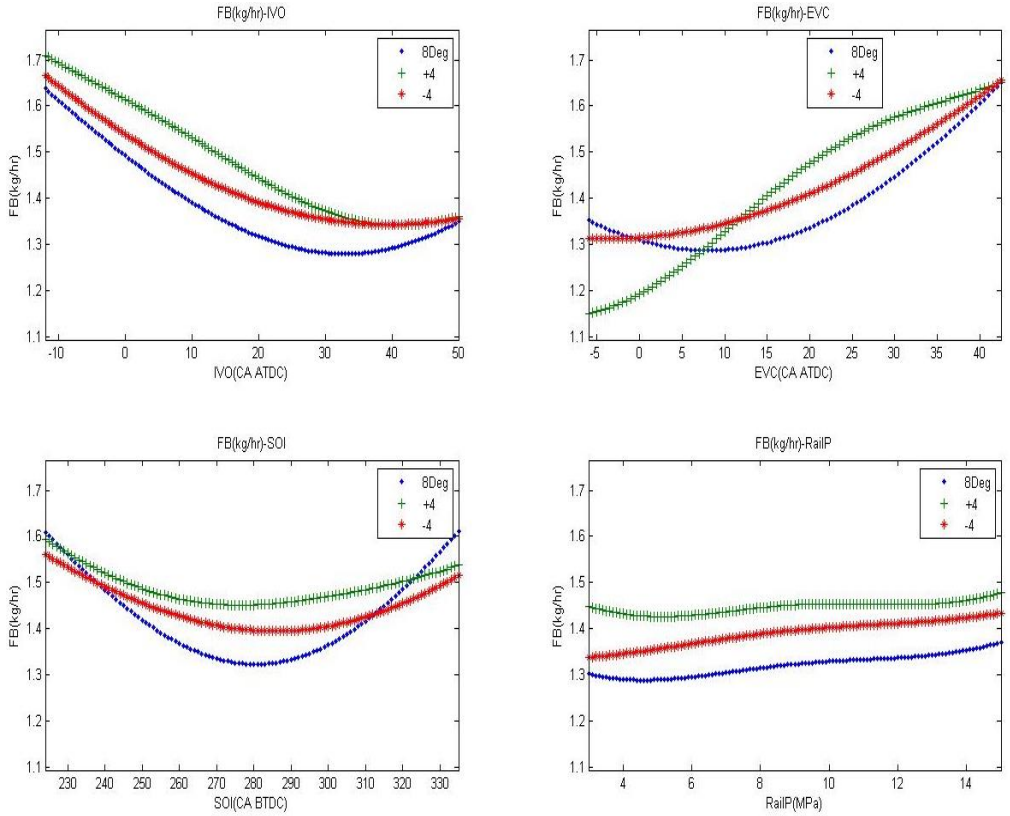


Figure 5.32: Effects of actuator settings on FC at 700 – 28 operating point, Nominal is showed by a blue line, retarded by red and advanced by green.

To sum up, the main conclusions made from analysis of the screening DoE test points are:

- Although Spark timing (ST) affects the engine performance (i.e. both fuel consumption and Pn) considerably, no significant benefit was seen in performing the DoE at different spark timings other than MFB50% spark time. So, for designing the mapping DoE, ST was fixed at the nominal setting MFB50%.
- Fuel rail pressure (FRP) design space reduced to 8< FRP <15, due to better engine performance (regarding Pn and fuel consumption) at all the operating points.
- Start of Injection (SOI) design space reduced to 260<SOI<330, where engine showed better performance at all the operating points.
- The boundary limits for IVO is -12 < IVO < 50, and for EVC is -6 < EVC < 44, however; in reality these two parameters cannot be changed independently for '700 (rpm) / 28 (Nm)' and '1500 (rpm) / 41 (Nm)' minimap points due to combustion stability issues. Therefore, the maximum overlap for the 700 – 28 operating point was limited to 10 deg, and for 1500 – 41 operating point was limited to 30 deg. Also, since at 700 – 28 operating point the engine performance was unstable when EVC > 30 deg ATDC, the design space for EVC was also reduced at this minimap point.

5.6.2.3 Mapping Stage: Implementation

The mapping DoE was planned using the sequential MB-MV DoE strategy. In order to validate the sequential DoE methodology, four MB-MV iterations were planned and run for the '2000 (rpm) / 81 (Nm)' minimap point as shown in Figure 5.33, i.e. using an OLH DoE with 50 test points for MB stage with subsequent model validation DoEs of size 15. Also, in order to maximize the utilization of the information gained from the screening experiments, the 'valid' screening test points (i.e. the test points that were inside the revised variable space after the screening stage) were carried over to the engine mapping stage. For instance, 15 valid test points were carried over from the screening

stage to the mapping stage for the ‘2000 (rpm) / 81 (Nm)’ minimap point. Therefore, the MB-MV DoE strategy for the ‘2000 (rpm) / 81 (Nm)’ minimap point was planned as following:

- **Stage 1** (MB-MV1): 65 test points for model building (MB1 = 15 ‘screening’ + 50 ‘MB’) and 15 test points for model validation (MV1), shown in Figure 5.34 (a).
- **Stage 2** (MB-MV2): 80 test points for model building (MB2 = MB1 +MV1) and 15 test points for model validation (MV2), shown in Figure 5.34 (b).
- **Stage 3** (MB-MV3): 95 test points for model building (MB3 = MB2 +MV2) and 15 test points for model validation (MV3), shown in Figure 5.34 (c).
- **Stage 4** (MB-MV4): 110 test points for model building (MB4 = MB3 +MV3) and 15 test points for model validation (MV4), shown in Figure 5.34 (d).

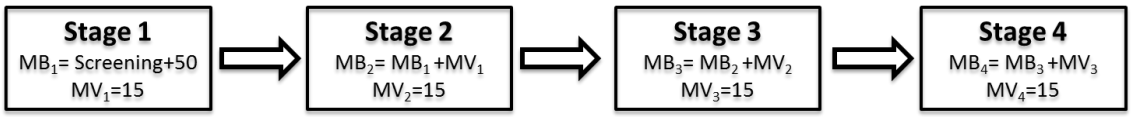
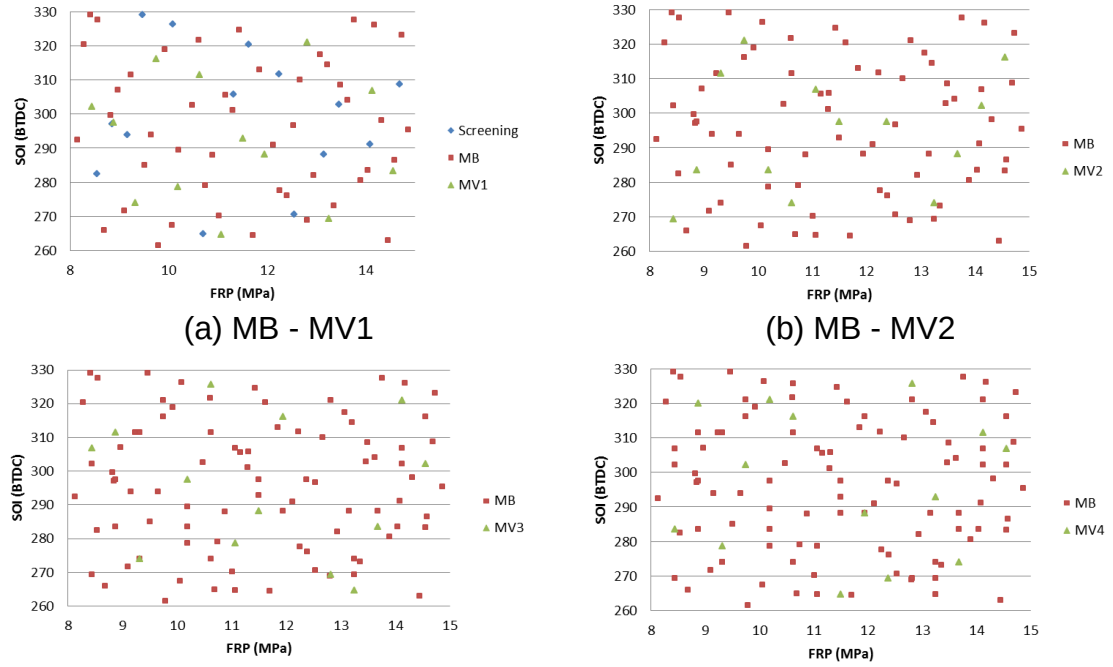


Figure 5.33: MB-MV DoE strategy planned for 2000-81 operating point

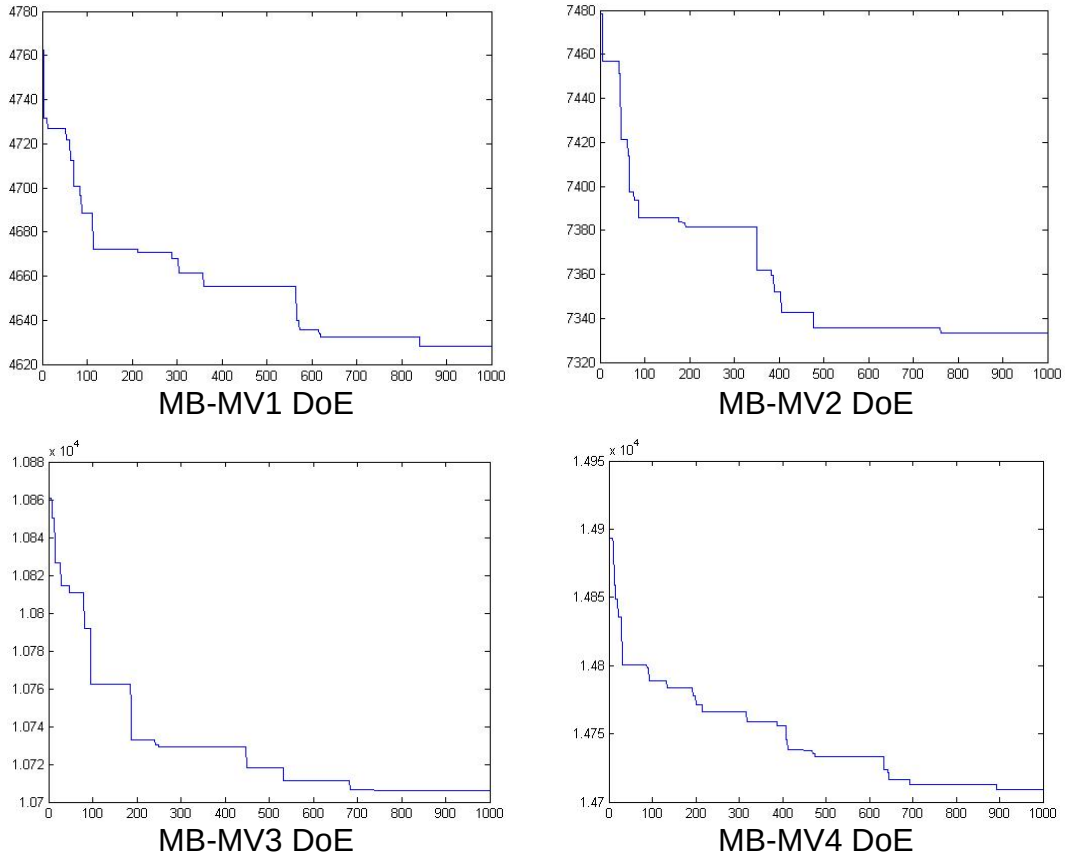
The DoE test points illustrated in Figure 5.34 were generated by running the developed PermGA, based on Equation 5.2, for 1,000 iterations, when the ‘Population Size’ was 200, ‘Crossover Rate’ was 0.8, ‘Mutation Rate’ was 0.05, and ‘Elite Size’ was 5 for MB DoE and 3 for MV DoEs. The PermGA convergence plot for each sequence of the MB-MV strategy is shown in Figure 5.35.



(a) MB - MV1 (b) MB - MV2

(c) MB - MV3 (d) MB - MV4

Figure 5.34: 2-D projection of mapping DoE test points for 2000 – 81 operating point in SOI – FRP coordinates



MB-MV1 DoE MB-MV2 DoE

MB-MV3 DoE MB-MV4 DoE

Figure 5.35: Convergence plot of PermGA for generating the MV DoEs

Figure 5.36 illustrates the Euclidian distance of the test points at each of the model validation stages, i.e. shown in Figure 5.34, using boxplots. This figure shows that variability of the Euclidian distances decreases by adding the subsequent MV DoEs, which in effect shows the capability of the developed PermGA algorithm to distribute the points evenly within the 4-dimensional design space even after 4 independent sequences of MV DoEs. Also, 4 outliers were seen at 'MB-MV4' DoE stage. The Euclidian distance of these outliers are not worse (less or more) than the previous DoE stage (i.e. 'MB-MV3'), however; since the variability is less at 'MB-MV4' (i.e. interquartile range is smaller) these test points are recognised as outliers. Moreover, Figure 5.37 characterises the distribution of Euclidian distances in terms of its standard deviation across the 4 MB-MV stages. This figure shows that the uniformity of the distribution of the test points within the design space is improving across the subsequent stages of MV DoEs.

The correlation (*r*) between each of the two DoE variables was also studied for all the 4 stages of the MB-MV strategy. It was observed that the correlation between the variables was negligible (i.e. $-0.05 \leq r \leq 0.05$) [90], thus, the designs are quasi-orthogonal.

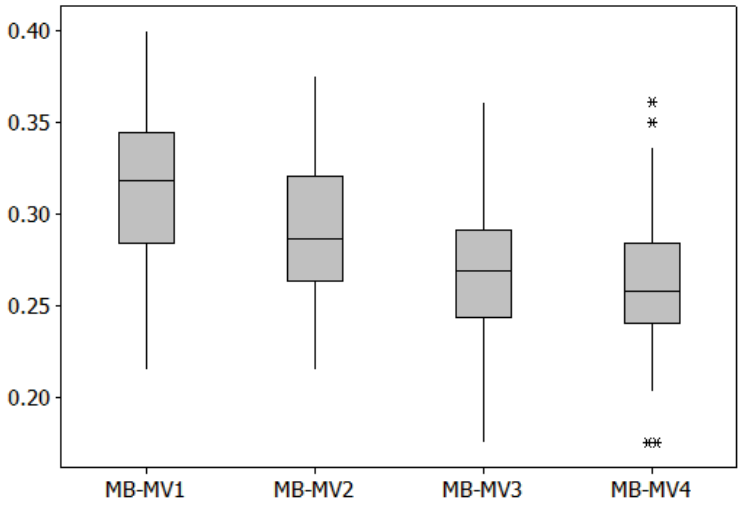


Figure 5.36: Boxplot of *Euclidean* distances across the subsequent validation DoEs

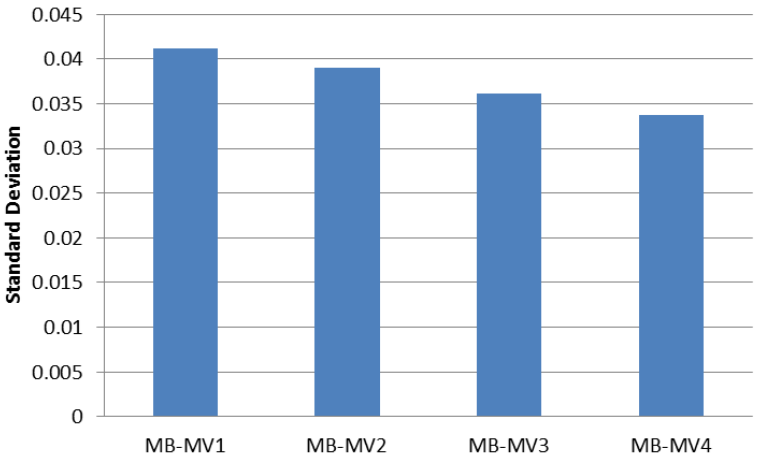


Figure 5.37: Standard deviation (σ) of the Mm Distances

Noteworthy, the MB-MV DoE strategy planned for the other 5 speed / load engine operating points was not in 4 stages. At these operating points, after carrying over the valid test points from the screening stage, an initial DoE of 50 points was generated for model building, and then it was augmented by the subsequent MV DoEs of 15 points until the target model accuracy was achieved.

5.6.2.4 Mapping Stage: Analysis of Results

After collecting engine responses of interest for the planned DoE test points at each stage of the MB-MV DoE strategy, a wide range of response models (e.g. polynomial, RBF and Hybrid RBF) were fitted using the MATLAB MBC Model Browser toolbox. The extreme observations (i.e. outliers) were eliminated from the model fitting process, where it was necessary. Subsequently, a response model was selected through evaluation of RMSE [156] and PRESS RMSE [224] statistical criteria. The quality of the selected response models was then judged based on analysing:

I. <u>Analysis of Residuals:</u>

Figure 5.38 and Figure 5.39 illustrate examples of the residual plots for the Pn and fuel consumption responses against the observation number (or versus time order in which the data was observed) at 2000 – 81 speed / load operating point. Residuals in these plots are the difference between the real response value (or collected value) and the predicted response value by the fitted model, for each MB test point. In general, it is expected to have (at least approximately)

random appearance for the residual plots. Observing trends in the pattern of residuals, either funnelling in / fanning out patterns which are indicators of decrease / increase in error variance, violates the constant-variance assumption [244]. It can be observed that both residual plots (Figure 5.38 and Figure 5.39) show random behaviour around the zero horizontal line, so, it is reasonable to believe that constant-variance assumption is valid for the fitted models.

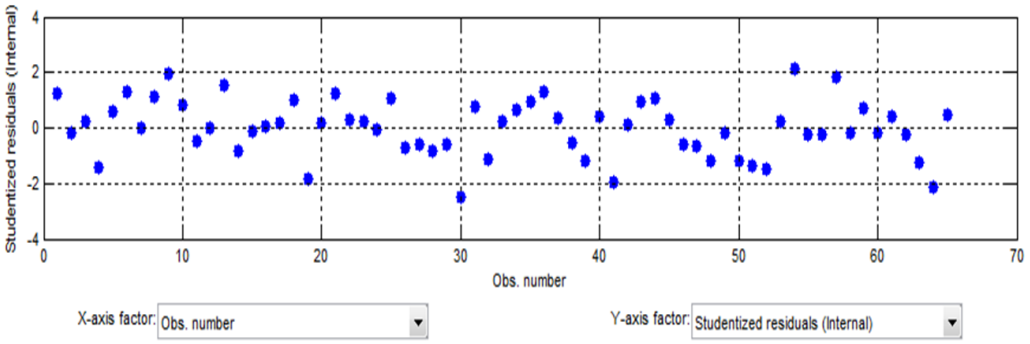


Figure 5.38: Residual plot for Pn response at 2000 – 81 operating point (using MBC Toolbox) after the first sequence (MB-MV1)

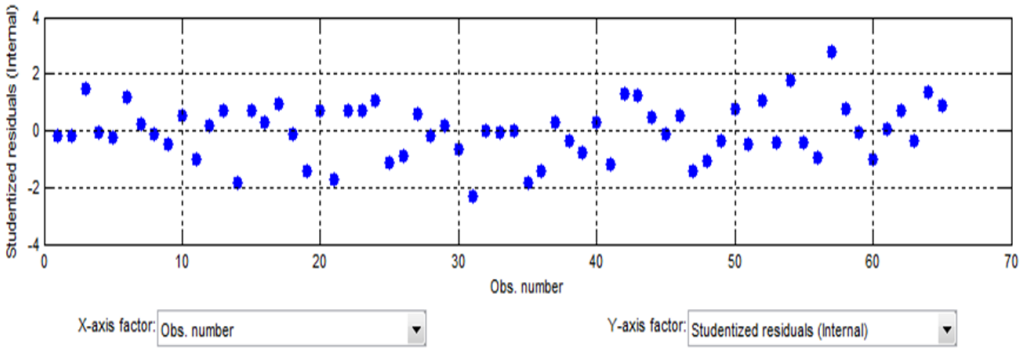


Figure 5.39: Residual plot for Fuel Consumption response at 2000 – 81 operating point (using MBC Toolbox) after the first sequence (MB-MV1)

II. Statistical diagnostics

Figure 5.40 and Figure 5.41 illustrate the improvement in model accuracy for the Pn and fuel consumption responses at different DoE stages, for the 2000 – 81 operating point. It can be seen that the PRESS RMSE [54] and Validation RMSE (see Equation 2.15) are decreasing over the subsequent stages of the sequential process, which indicates that the quality of the response models is improving sequentially.

Similarly, Figure 5.42 and Figure 5.43 illustrate the improvement in model accuracy of Pn and fuel consumption responses at the 2000 – 81 minimap point, in terms of the model prediction relative error, which is expressed as the ratio of validation RMSE to mean response as percentage (see Equation 2.16). The decreasing trend in the validation RMSE shows that the quality of the response surfaces is enhanced step by step. It can be seen that for the 2000 – 81 minimap point illustrated in Figure 5.42 and Figure 5.43, the relative error is 0.91 % for fuel Consumption and 8.4 % for Pn after the 4th MB-MV iteration. Given the engineering target for model quality of FC and Pn responses at 2000 – 81 minimap point (1% and 10%, respectively), it could be argued that for this case the engine response models were acceptable after the second MB-MV iteration, i.e. based on a mapping DoE of only 80 (65 MB + 15 MV) test points. This is significantly less than the normal mapping DoEs which typically use 120-150 test points [43], with no automatic guarantee that the target model accuracy will be reached.

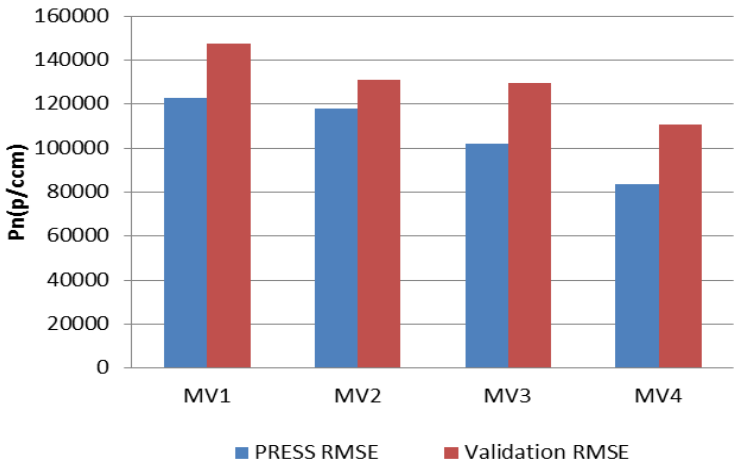


Figure 5.40: PRESS RMSE and Validation RMSE for Pn response at different sequences at 2000 – 81 operating point

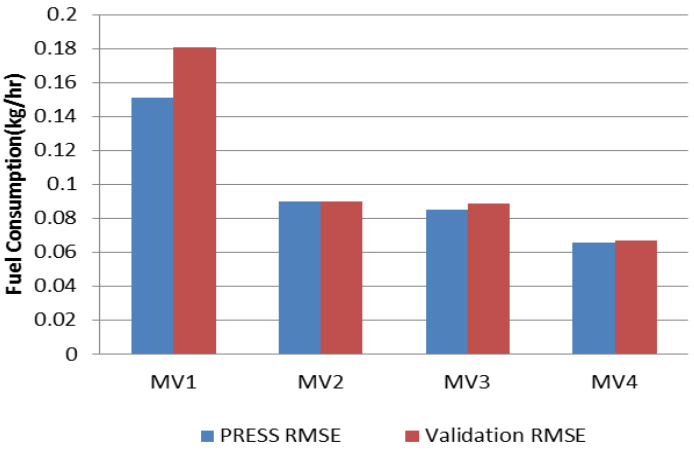


Figure 5.41: PRESS RMSE and Validation RMSE for fuel consumption response at different sequences at 2000 – 81 operating point

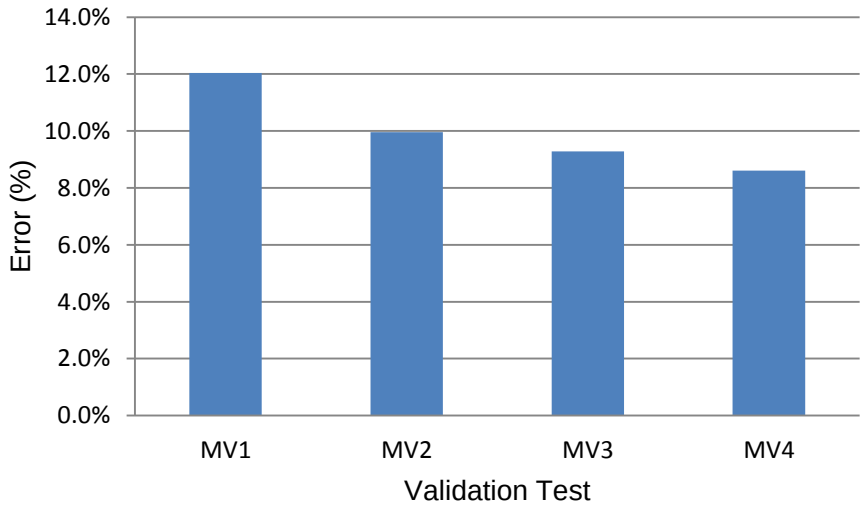


Figure 5.42: Pn prediction relative error at 2000 – 81 operating point

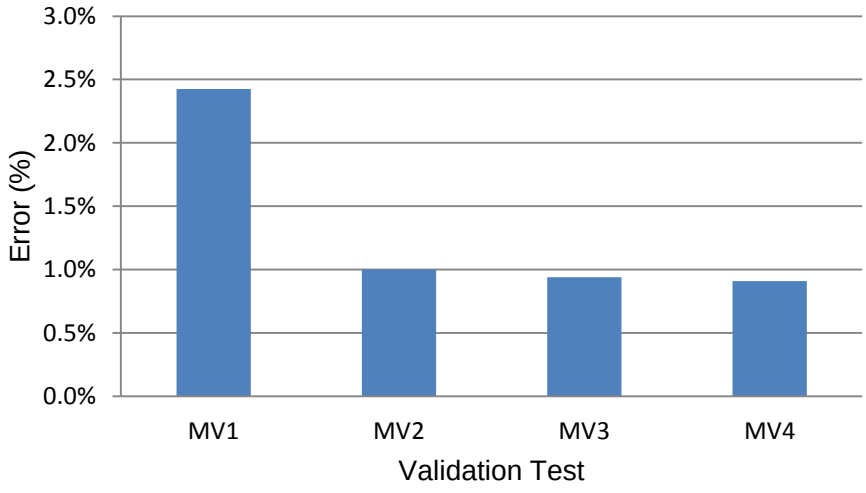


Figure 5.43: Fuel consumption prediction relative error at 2000 – 81 point

III. Trend Analysis

Figure 5.44 to Figure 5.47 illustrate plots of Pn and fuel consumption responses through stages MV1 and MV4. These figures clarify how the shape and trend of the responses, particularly the fuel consumption response, are transformed iteratively through collecting more infill test points, improving the prediction accuracy throughout the design space. As an example, it can be seen that the shape of fuel consumption response after collecting 4 sets of validation points (Figure 5.47) is significantly different from the response model at sequence 1 (Figure 5.46), especially at the extremities of the design space, i.e. the corner areas.

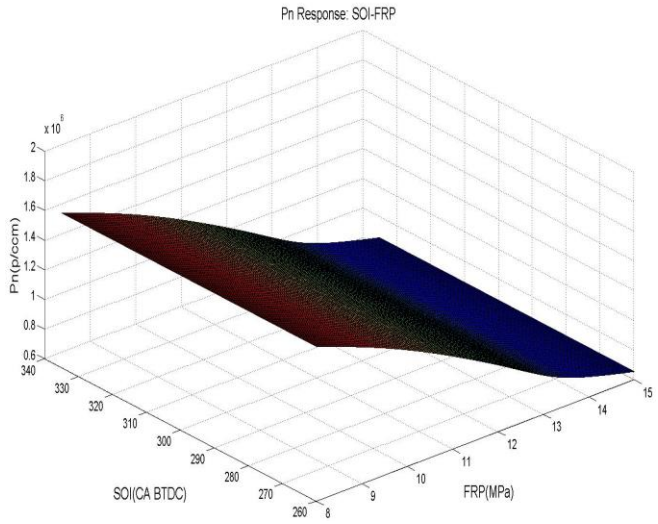


Figure 5.44: Pn response at MV1 stage at 2000 – 81 operating point

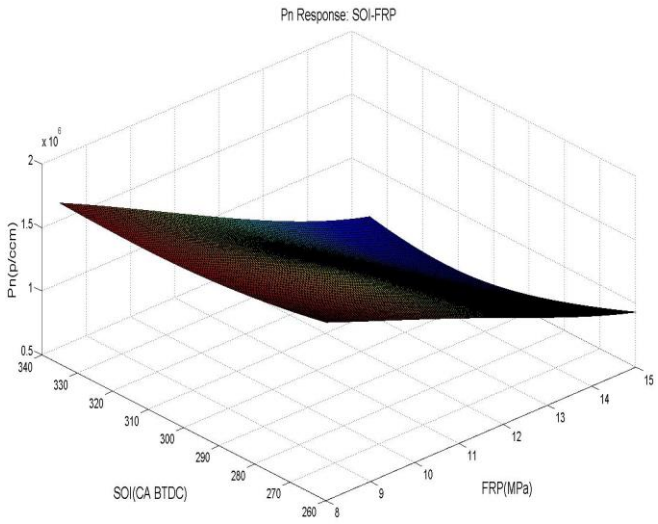


Figure 5.45: Pn response at MV4 stage at 2000 – 81 operating point

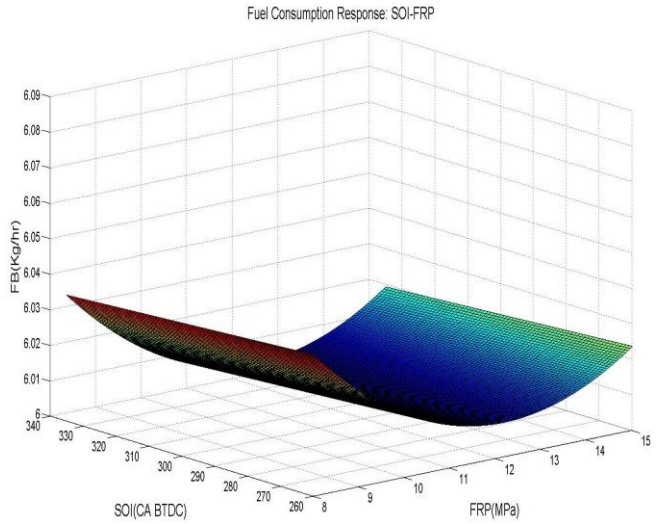


Figure 5.46: Fuel consumption response at MV1 stage at 2000 – 81 operating point

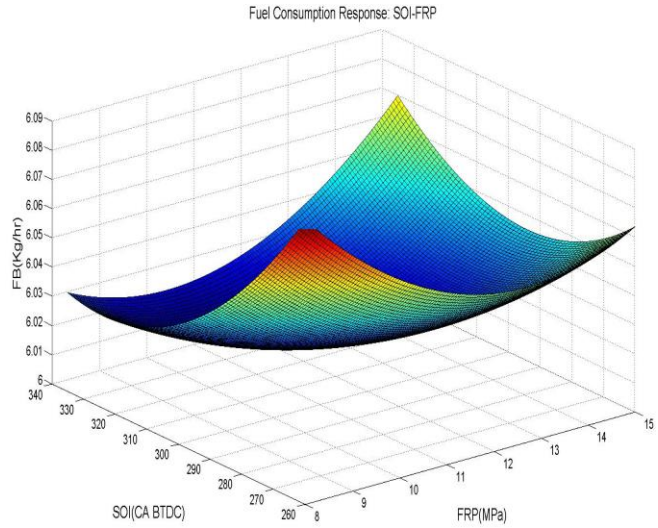


Figure 5.47: Fuel consumption response at MV4 stage at 2000 – 81 operating point

The last stage of engine model validation was based on the engineering analysis of the fitted models to ensure that the models are not over-fitted or extrapolating among the collected data. Figure 5.48 and Figure 5.49 illustrate the Pn and Fuel consumption response surfaces at 2000-81 operating point after four stages of sequential data collection.

Figure 5.48 shows that for low Fuel Rail Pressure (FRP) and late start of injection (SOI), shown as Area 1, the fuel consumption is higher due to poor mixture preparation and loss of efficiency. It can be seen that fuel consumption decreases with higher FRP (Area 2), which could be linked to improved fuel atomization and better fuel mixture preparation. Similarly, Figure 5.49 illustrates

the Pn response against FRP-SOI. For low FRP and late SOI, area 1, the Pn emissions are relatively high, which is due to insufficient time for mixture preparation (inhomogeneous charge at ignition). Pn is decreasing with increasing the FRP, moving from area 1 toward area 2, which is due to better fuel atomization. However, the Pn is high at low FRP and early SOI, area 3, which could be linked to locally fuel rich zones as a result of poor atomization or fuel impingement on the piston. Furthermore, it is seen that Pn is decreasing with increasing FRP (area 4), due to improved fuel atomization resulting in a more homogenous mixture.

Similar engineering analysis was carried out for engine response models at each engine speed / load operating point tested to validate the observed trends.

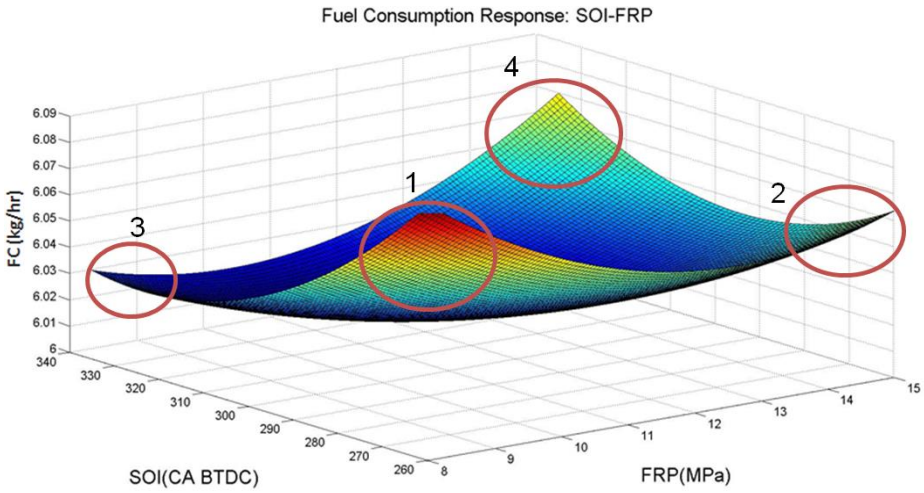


Figure 5.48: Fuel consumption response in SOI-FRP coordination at stage 4

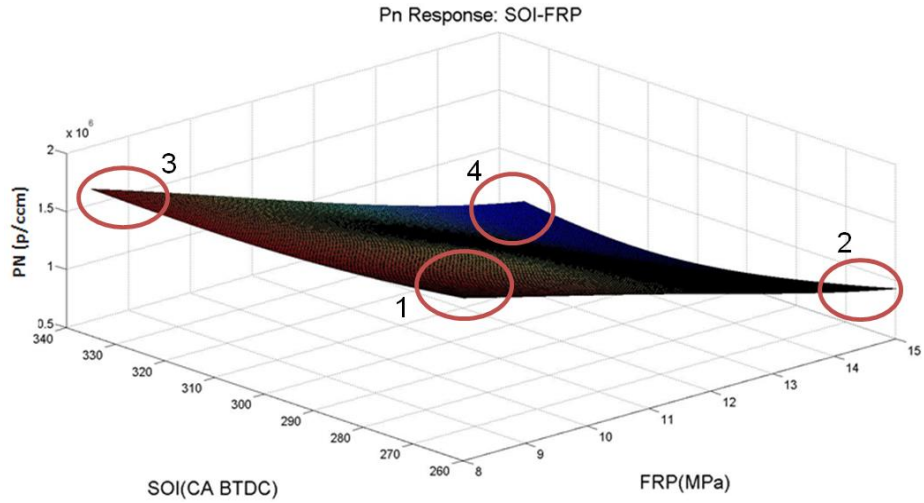


Figure 5.49: Pn response in SOI-FRP coordination at stage 4

Table 5.5 summarises statistical properties of the fitted approximation models for the AJ133 GDI engine key responses (i.e. fuel consumption and Pn). In this table, the 'MB Test Points' column indicates the number of feasible MB test points used to fit the high fidelity response models after removing the outliers. Noteworthy, only the MB-MV DoE strategy for minimap 5 (i.e. 2000 – 81 operating point) was planned in 4 stages, to validate the sequential MB-MV DoE methodology. The sequential data collection process for the other minimap points was stopped when the response models of target accuracy were achieved.

Minimap	Response	Model Type	MB Test Points	RMSE	PRESS	Relative Error (%)
1	FC	RBF-thinplate	69	0.011	0.013	0.93
	Pn	RBF-thinplate	71	2.98E+3	4.61E+3	29.47
2	FC	RBF-thinplate	81	0.025	0.027	0.89
	Pn	Quadratic-RBF	83	2.50E+3	4.43E+3	28.26
3	FC	Quadratic-RBF	57	0.014	0.018	0.61
	Pn	RBF-thinplate	59	9.74E+4	1.11E+5	9.89
4	FC	Quadratic-RBF	60	0.015	0.017	0.58
	Pn	RBF-thinplate	63	8.48E+4	1.10E+5	6.89
5	FC	RBF-thinplate	110	0.060	0.063	0.91
	Pn	Quadratic-RBF	109	9.16E+4	8.29E+4	8.40
6	FC	Quadratic-RBF	65	0.022	0.025	0.46
	Pn	Quadratic-RBF	65	1.63E+5	2.24E+5	7.34

Table 5.5: Summary of fitted MBC models for GDI engine with single injection strategy

5.6.3 Double-Injection Phase

5.6.3.1 Screening Stage DoE: Implementation

Similar DoE strategy was taken to conduct the screening DoE stage in the double-injection phase. Accordingly, an OLH DoE of 60 test points was generated based on the calibration variables summarised in Table 5.6. In the double-injection phase, spark timing was fixed on the nominal value, i.e. MFB50% position at 8°ATDC.

Code	*Name*	*Range*		*Unit*
SOI1	Start of Injection 1	180	360	deg BTDC
Delay	Delay between injection 1 and injection 2	1	8	ms
Ratio	Proportion of total fuel injected in injection 1	30	70	%
FRP	Fuel Rail Pressure	8	15	MPa
IVO	Inlet Valve Opening	-12	50	deg ATDC
EVC	Exhaust Valve Closing	-6	44	deg ATDC

Table 5.6: Engine control variables for the GDI engine with double injection strategy

The distribution of the 60 screening test points within the 6-dimensional design space is shown in Figure 5.50. This DoE was generated by running the developed PermGA, based on Equation 5.1, for 1,000 iterations, when the 'Population Size' was 200, 'Crossover Rate' was 0.8, 'Mutation Rate' was 0.05, and 'Elite Size' was 6. The PermGA convergence plot is shown in Figure 5.51. To illustrate uniformity of the distributed test points within the 6-dimensional design space, the Euclidean minimum distance, i.e. with normalised parameters between [0 1], was illustrated in Figure 5.52 for all the generated test points. Accordingly, it can be argued that the PermGA algorithm could deliver a 6-dimentional OLH design with a good space filling property.

The correlation between each of the two variables (out of 6) was also studied. The maximum correlation among the variables is (- 0.07), between variables Delay and Ratio. So, it can be concluded that the generated 6-dimensional OLH design is quasi-orthogonal, since the correlation is negligible [90].

As discussed in Chapter 4, the double injections strategy could not be used for low speed / load minimaps, i.e. '700 – 28' and '1500 – 41' speed/load operating points. It was due to the ECU restriction on minimum pulse width, which defines the minimum fuel injection required at each pulse. So, the screening and mapping DoEs in the double-injection phase were carried out for the 4 higher speed / load operating points, within the University of Bradford dynamometer test range, as detailed in Table 5.7.

Test Point	Engine Speed [rpm]	Engine Load [Nm]
3	1250	125
4	1500	105
5	2000	81
6	2000	199

Table 5.7: Screening operating points

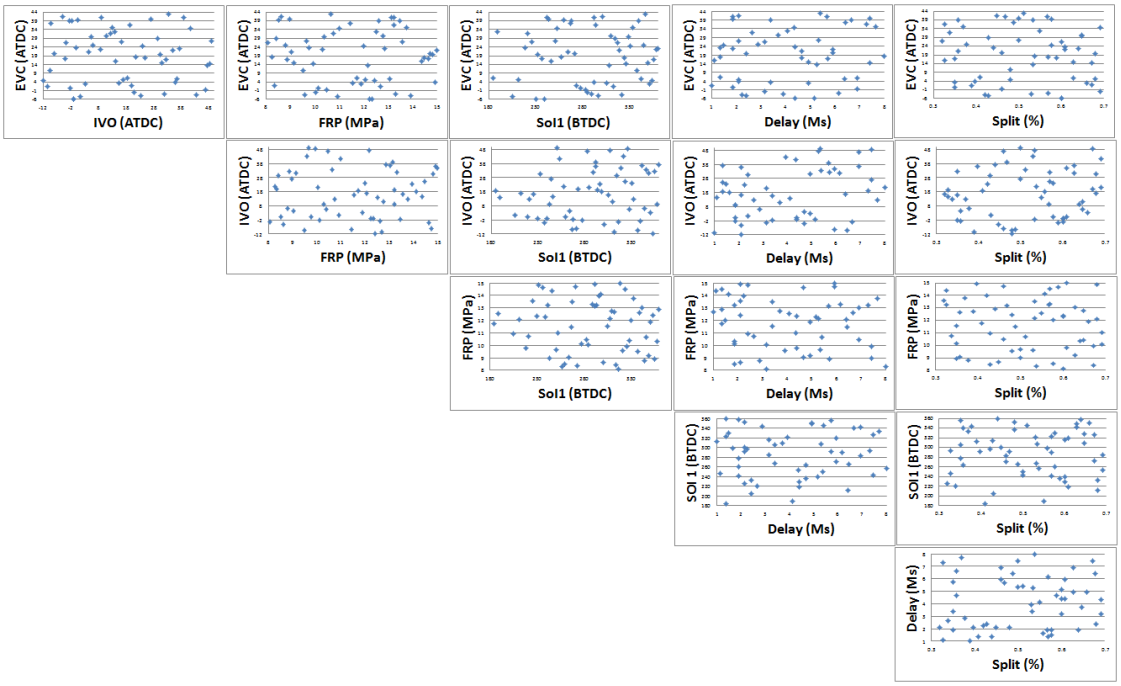


Figure 5.50: Distribution of 60 Screening DoE test points in design space

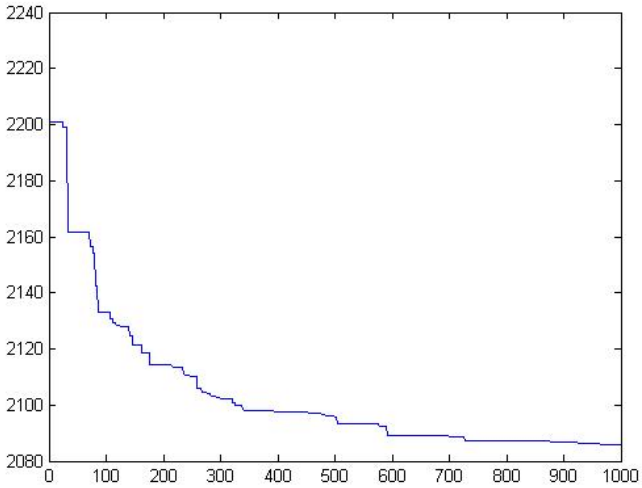


Figure 5.51: PermGA Convergence plot for generating the Screening DoE

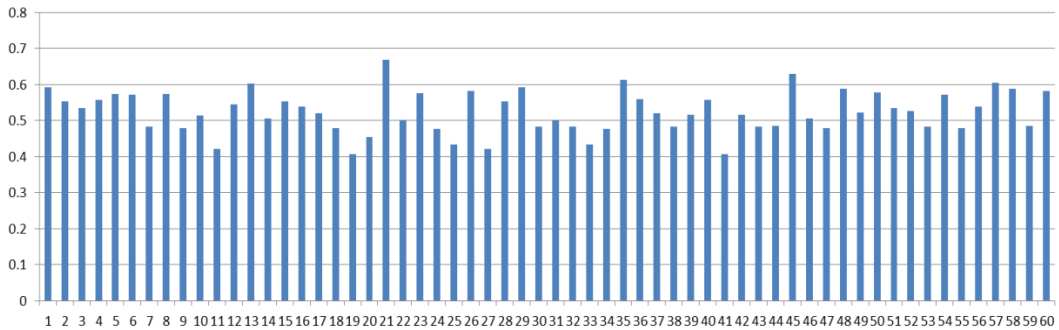


Figure 5.52: Euclidean minimum distance for the Screening DoE test points

5.6.3.2 Screening Stage DoE: Analysis of Results

After collecting the required information at the designed screening test points, the collected data was analysed as following:

I. Combustion Stability Analysis:

A feasibility study was executed on the collected data to identify the points that do not meet the combustion stability set threshold. Unlike the single injection phase, most of the screening test points were feasible. However, the 1500 – 105 screening DoE had a number of unstable test points. Figure 5.53 illustrates the distribution of feasible and infeasible test points for the 1500 (rpm) / 105 (Nm) operating point, in 'EVC-IVO' coordinates. The graph shows that late EVC has a detrimental effect on combustion stability, particularly when the IVO is early (high overlap). This can be seen more clearly by considering the 'Overlap' between EVC and IVO, i.e. plotting (EVC-IVO) versus EVC, as illustrated in Figure 5.54. This shows that at the 1500 (rpm) / 105 (Nm) operating condition, overlap (EVC-IVO) higher than 30 degrees has negative effects on combustion stability.

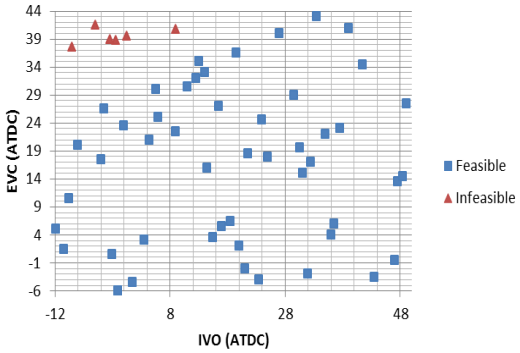


Figure 5.53: Distribution of infeasible test points in EVC – IVO coordinates

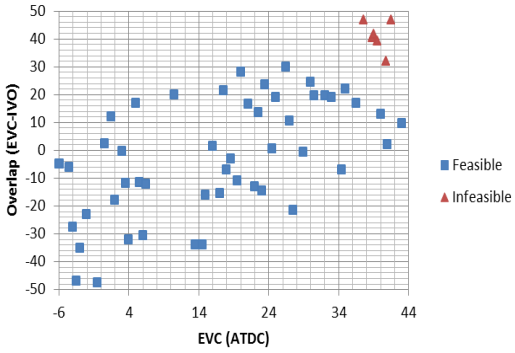


Figure 5.54: Distribution of infeasible test points in Overlap – EVC coordinates

II. Engine Response Surface Models:

Different response surface models were fitted based on the 'valid' screening test points using the MATLAB MBC toolbox. It was seen that RBF models with thinplate kernel showed better statistical properties than the other candidate models. Thus, RBF-thinplate was used to build the response models of interest.

Figure 5.55 and Figure 5.56 show an example of the fitted response models for fuel consumption and Pn responses at 1500 – 105 operating point, respectively.

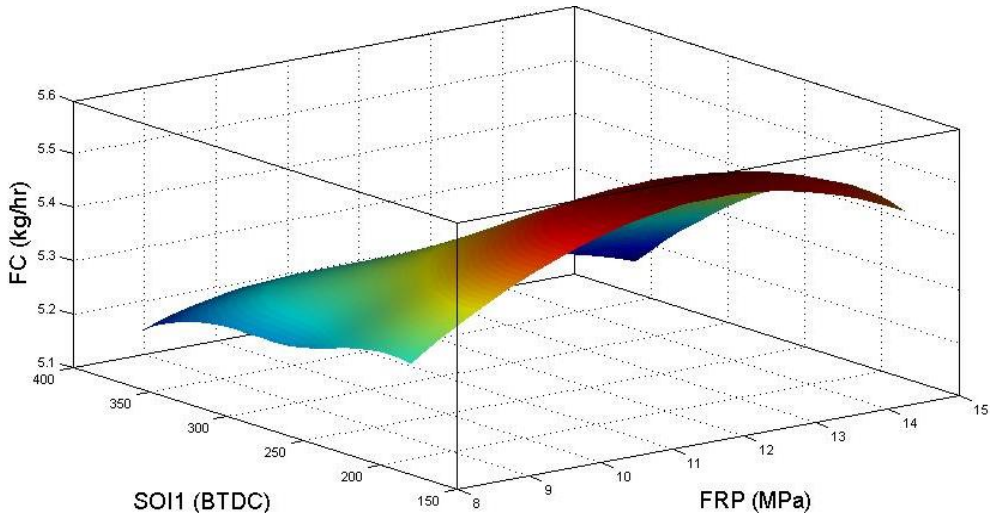


Figure 5.55: 3D View of FC versus SOI1 and FRP at 1500 – 105 minimap point

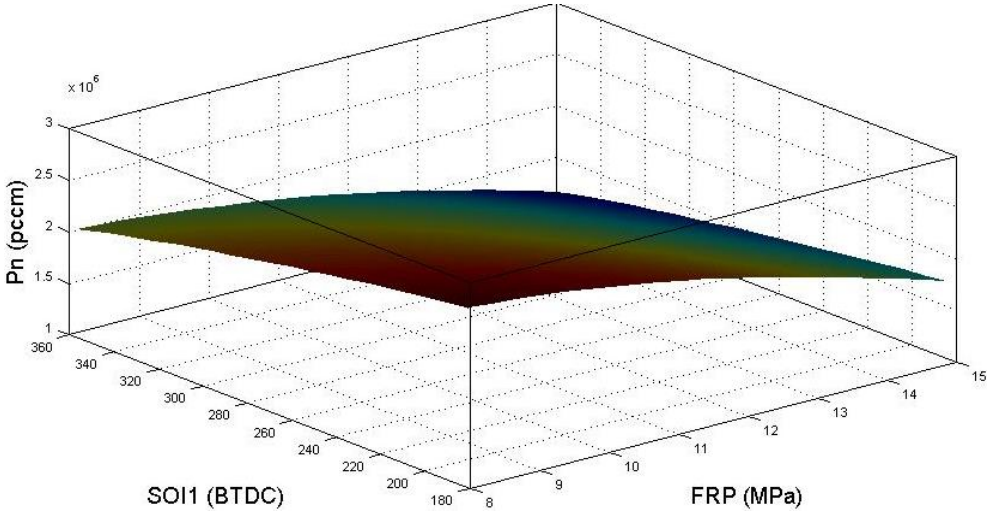
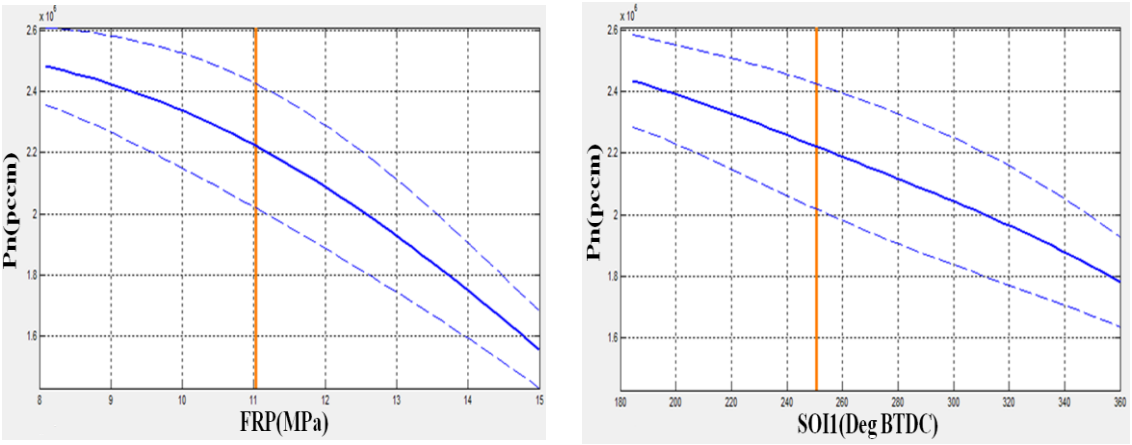


Figure 5.56: 3D View of Pn versus SOI1 and FRP at 1500 – 105 minimap point

III. Trend Analysis:

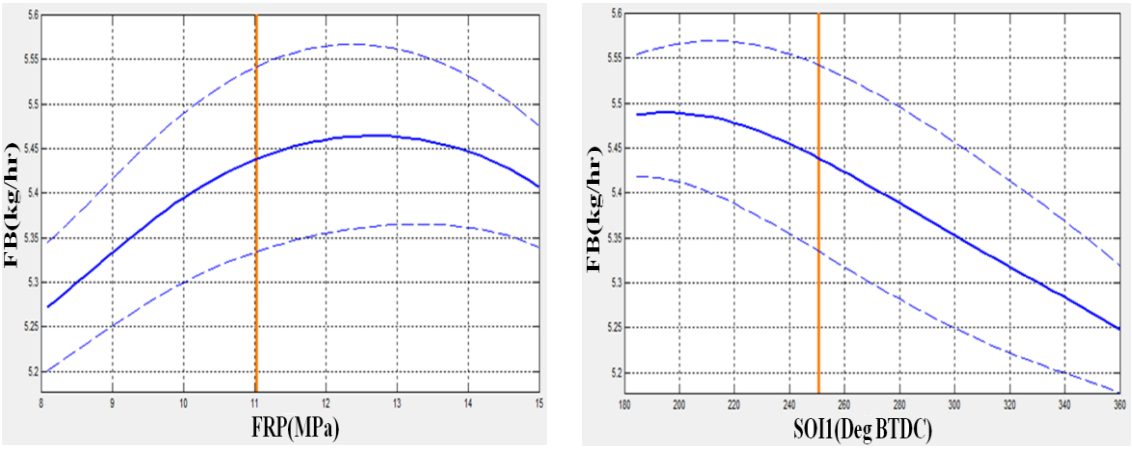
Figure 5.57 and Figure 5.58 illustrate examples of pseudo 'main effects' type plots generated for the fitted RBF models at the 1500 rpm / 105 Nm minimap point. These figures are plotted using the MATLAB Model Based Calibration toolbox. These figures show that both fuel consumption and Pn responses are improved at the higher end of the FRP range. This enhancement, especially in Pn, is due to better atomization of fuel in higher FRP and consequently a better fuel mixture preparation. Therefore, the FRP design space could be reduced for

the engine mapping stage. Similarly, a trend analysis was conducted on the SOI1 parameter. It was observed that the higher range of the SOI1 variable, corresponding to 'early' start of injection, is beneficial for both fuel consumption and Pn. The main reason for this enhancement is the further time for mixture preparation when the start of injection is early. Worth mentioning, similar trends were also observed for the other tested minimap points.



(a) (b)

Figure 5.57: Pn against (a) FRP(MPa), and (b) SOI1(BTDC), within 95% confidence intervals



(a) (b)

Figure 5.58: Fuel consumption against (a) FRP(MPa), and (b) SOI1(BTDC), within 95% confidence intervals

The data analysis was executed on the collected data at all the 4 minimap points tested for the GDI engine with double injection strategy. All the results were submitted to the CREO project as a technical report 'CREO-15032013: Preliminary Report on Phase 2 Screening DoE Results for Multiple Injections' [245].

To summarise, the main conclusions made from analysing the screening test points are:

- The FRP range reduced to: $11\ MPa \leq FRP \leq 15\ MPa$;
- The SOI1 range reduced to: $240\ BTDC \leq SOI1 \leq 360\ BTDC$;
- At 1500 – 105 operating point, the maximum valve overlap was constrained to 30 deg.

5.6.3.3 Mapping Stage: Implementation

Similar to the single injection phase, the MB-MV sequential DoE strategy was used to conduct the mapping DoE stage. Accordingly, after carrying over the valid test points from the screening stage, an initial OLH DoE of 50 points was generated for model building. This design was then augmented by the subsequent MV OLH DoEs of 15 points, until the target model accuracy was achieved.

For illustration, the first iteration of MB-MV DoE for the '2000 (rpm) / 81 (Nm)' minimap point is shown in Figure 5.59. This figure shows the location of 25 'valid' screening test points, 50 model building test points, and 15 model validation points in SOI1-FRP coordinates, at the first sequence of the mapping strategy. The DoE test points illustrated in Figure 5.59 were generated by running the developed PermGA, based on Equation 5.2, for 1,000 iterations, when the 'Population Size' was 200, 'Crossover Rate' was 0.8, 'Mutation Rate' was 0.05, and 'Elite Size' was 5 for MB DoE and 3 for MV DoE. The PermGA convergence plot for generating the MV DoE is shown in Figure 5.60.

The *Euclidean* minimum distance, i.e. with normalised parameters between [0 1], for the generated DoE test points (i.e. 50 MB + 15 MV test points) within the 6-dimensional design space is illustrated in Figure 5.61. This histogram indicates that distribution of the generated test points within the 6-dimensional design space is quasi-uniform.

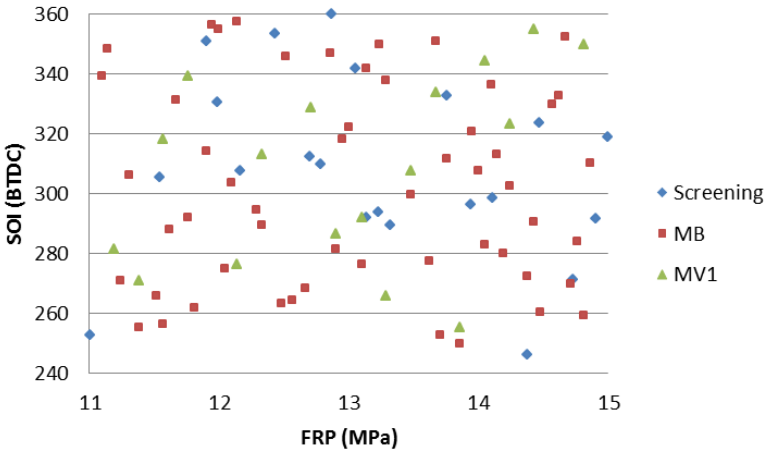


Figure 5.59: MB-MV first sequence at 2000 – 81 operating point

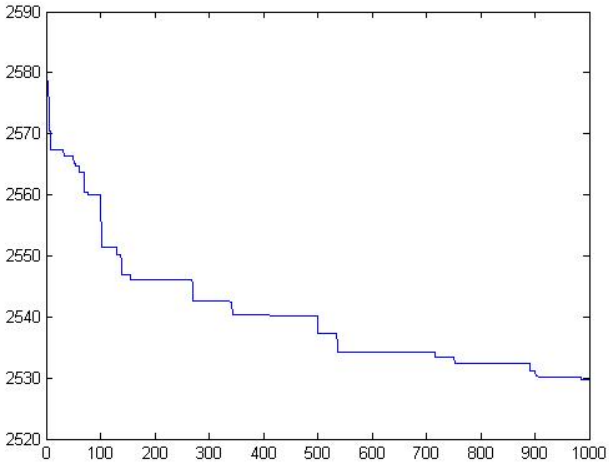


Figure 5.60: Convergence plot of PermGA for generating the MV1 DoE

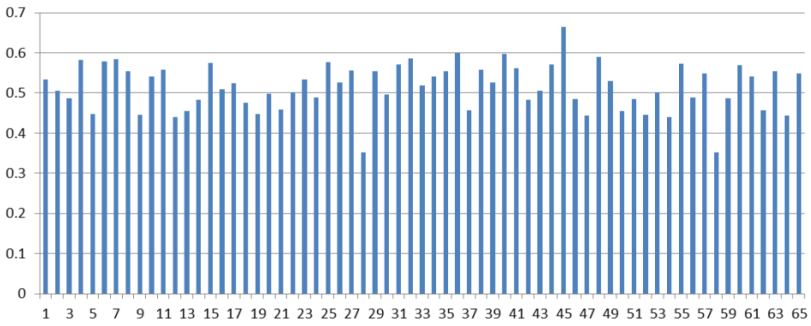


Figure 5.61: Euclidean distance for the MB-MV1 DoE at 2000 – 81 point

The uniformity of distribution of the test points is also illustrated using boxplots, as shown in Figure 5.62. It can be seen that the variability of the Euclidian distances decreased by adding the subsequent MV DoE. Therefore, the

developed PermGA could distribute the validation test points evenly within the 6-dimensional design space.

The correlation (*r*) between each of the two variables (out of 6) was also studied for the MB-MV1 DoE. It was observed that the correlation between the variables was insignificant (i.e. $-0.08 \leq r \leq 0.08$) [90], which means that the MB-MV1 DoE is quasi-orthogonal.

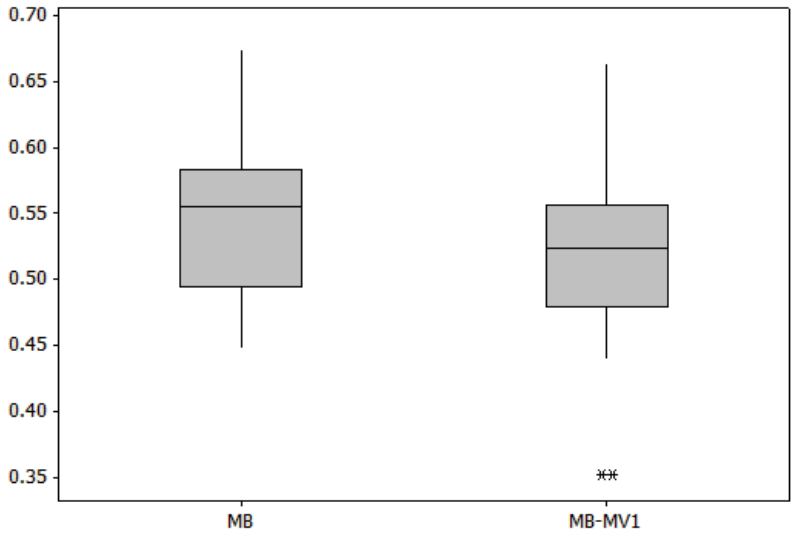


Figure 5.62: Boxplot of *Euclidean* distances after the first validation DoE

5.6.3.4 Mapping Stage: Analysis of Results

A modelling process similar to the single injection phase was used to fit the required engine response models. Afterwards, the quality of the fitted response models was judged based on analysing:

I. <u>Analysis of Residuals:</u>

Figure 5.63 and Figure 5.64 illustrate the residual plots of the final Pn and fuel consumption responses at the 2000-81 operating point. It can be seen that both plots show random behaviour around the zero horizontal band, so; it can be concluded that the constant-variance assumption is valid for the fitted models. In other words, it is expected that no additional variables, than the already considered parameters in Table 5.6, to be effective on the responses.

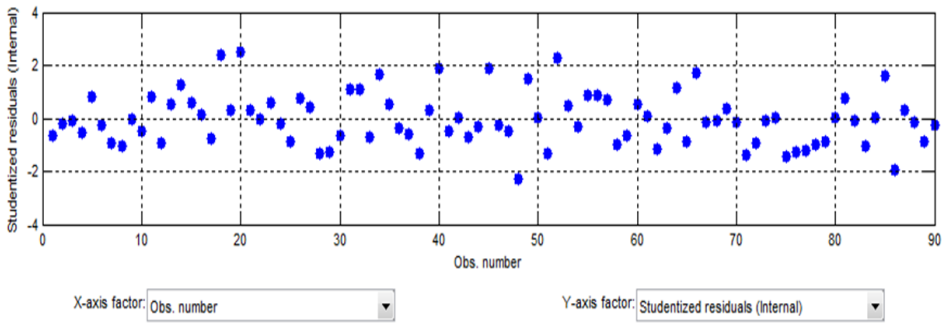


Figure 5.63: Residual plot for the final Pn response at 2000-81 operating point (using MBC Toolbox)

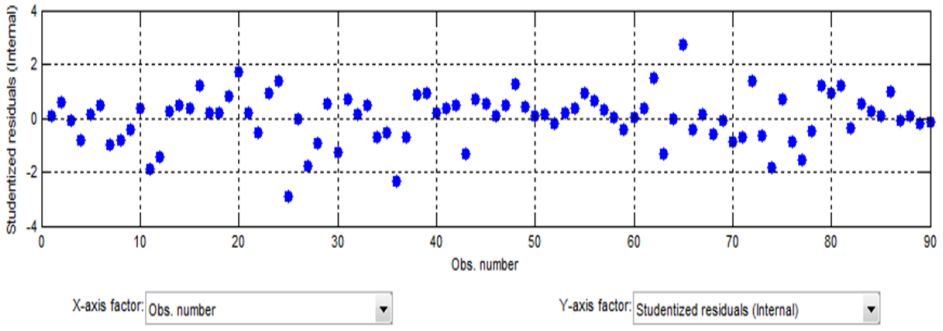


Figure 5.64: Residual plot for the final fuel consumption response at 2000-81 operating point (using MBC Toolbox)

II. Statistical diagnostics

Table 5.8 summarises statistical properties of the final MBC models fitted for the Pn and FC responses, including RMSE, PRESS RMSE and relative error measure. Similar to the single injection phase, the target model accuracy was achieved using less test points than the conventional one-shot DoE approaches (i.e. usually 120-150 test points).

Minimap	Response	Model Type	MB Test Points	RMSE	PRESS	Relative Error (%)
3	FC	Quadratic-RBF	91	0.016	0.019	0.82
	Pn	RBF-linearrbf	91	3.71E+4	4.70E+4	7.10
4	FC	Quadratic-RBF	92	0.014	0.017	0.76
	Pn	Quadratic-RBF	92	3.37E+4	4.43E+4	7.18
5	FC	Quadratic-RBF	90	0.014	0.020	0.81
	Pn	Quadratic-RBF	90	3.59E+4	5.42E+4	6.13
6	FC	RBF-thinplate	90	0.039	0.046	0.73
	Pn	RBF-thinplate	90	5.43E+4	8.28E+4	6.61

Table 5.8: Summary of fitted MBC models for GDI engine with double injection strategy

III. Trend Analysis

The last stage of model validation was to compare the model trends with the engine physical behaviour. Figure 5.65 and Figure 5.66 illustrate the Pn and fuel consumption responses at the 2000 – 81 operating point against the FRP-SOI coordinates. It can be seen that these two figures show similar trends to the described engine models at the single-injection phase, shown in Figure 5.48 and Figure 5.49, which is consistent with the engineering expectations at this engine operating condition.

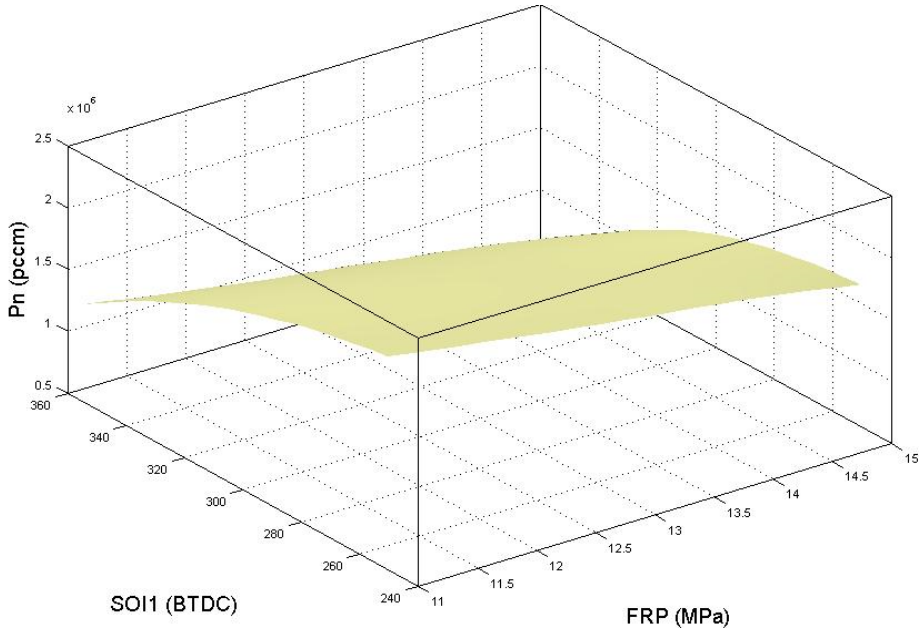


Figure 5.65: Pn response in SOI1-FRP coordination for 2000 – 81 minimap

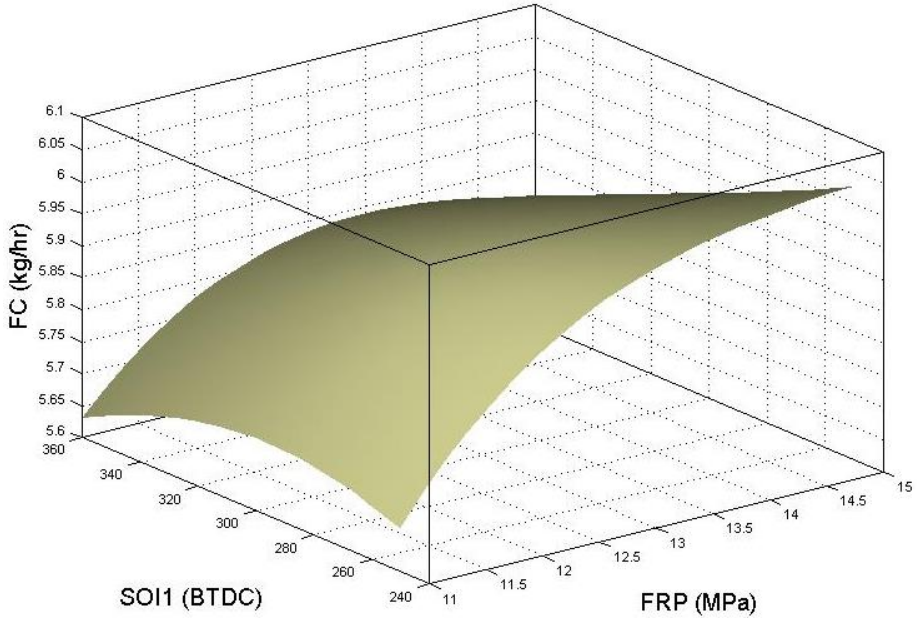


Figure 5.66: Fuel response in SOI1-FRP coordination for 2000 – 81 minimap

5.7 Summary, Discussion and Conclusions

The main aim of this chapter was to develop a sequential DoE strategy to enhance the effectiveness of the engine steady state data collection process, to maximise the information gained with minimal resource expenditure, in terms of engine testing. The proposed strategy offers a coherent framework for the Screening – Model Building – Model Validation sequence, based on an Optimal Latin Hypercube DoE. Accordingly, a Permutation Genetic Algorithm (PermGA) was developed and implemented in MATLAB to generate the OLH designs according to the AELH optimality criteria. A separate algorithm, also based on PermGA, was developed to generate optimal infill strategies for the model building – model validation sequence. This algorithm is then modified using a penalty function to be employed efficiently for problems with asymmetric design spaces. A validation exercise was carried out based on two test cases: one based on a theoretical function, to investigate the statistical properties of the experimental design, and one based on the AJ133 GDI engine mapping experiment. The GDI engine case study illustrated the application of the OLH DoE in the screening stage, aimed at identifying the useful calibration variables space, followed by a mapping series of experiments based on a MB-MV sequentially augmented OLH DoEs.

The main innovative feature of the sequential DoE framework for steady state engine mapping discussed in this chapter consists of the fact that it combines the ‘screening’ phase with the model building phase and a flexible sequence of MB-MV DoEs. This not only enhances the effectiveness of the engine data collection process, but it offers a much richer screening process. By fitting models based on the screening DoE data, the actuator range is not defined only by the feasible range, as it is the case with current approaches used in practice and discussed in literature [241], [246], but the analysis of the trends (including significant effects of interactions between control variables) allows the scope for the mapping experiments to be further narrowed down to areas of interest for the optimisation. A smaller variable space enables a better exploration through a smaller size mapping DoE experiment, hence an increased effectiveness of the test plan.

The main conclusions of applying the sequential MB-MV strategy can be summarised as:

1) PermGA and Infill PermGA algorithms can successfully generate uniformly distributed space filing designs for both symmetric and asymmetric design spaces. These algorithms can also maintain the experimental design orthogonality.
2) The accuracy of the engine response models is improving through the MB-MV sequence.
3) The accuracy of the response surfaces can be monitored at each stage, thus, the data collection process can be stopped when the required model accuracy is achieved.
4) MB-MV strategy has the potential to reduce both testing and computational effort.
5) MB-MV strategy is performing well regardless of problem dimensionality and variables' design space.

It is noteworthy that the practical implementation of this methodology for engine mapping and calibration studies substantially relies on the stable operation of the engine test facility, as engine characterization DoE data for each minimap point is not collected in a single sequence, which is the common industry practice.

In the following chapter, the developed high fidelity response models for the GDI engine case study are used for the calibration optimisation purposes.

Chapter 6: Evaluation of Multi-Disciplinary Optimisation Frameworks for Engine calibration Optimisation

6.1 Introduction

This chapter presents research to enhance the steady state engine calibration optimisation process with the application of multidisciplinary design optimisation frameworks. The research was carried out within the following steps:

1. Implementation of the conventional 2-stage optimisation approach for the GDI engine case study using the engine response models fitted in Chapter 5.
2. Application of 3 MDO frameworks (i.e. MDF, CO and ATC) to formulate and solve the GDI engine calibration optimisation problem, using the engine response models fitted in Chapter 5.
3. Validation of the developed MDO frameworks on the diesel engine case study.
4. Comprehensive discussion of the results and making conclusions.

6.2 Review of Conventional 2-Stage Calibration Optimisation Process for the GDI Engine Steady State Calibration Optimisation Problem

The conventional 2-stage calibration optimisation approach consists of two optimisation stages:

1- **'Local level' optimisation** (i.e. at each minimap point): In this level of optimisation, the main target is to study the trade-off between the conflicting key objectives at each minimap point, to find a number of candidate calibration settings at each engine speed / load operating condition [7], [209].

2- **'Global level' optimisation** (i.e. over all the minimaps or a reference driving cycle): In this optimisation level, the main objective is to select the best combination of local candidates which can satisfy the global requirements [210].

6.2.1 Local Level Optimisation

6.2.1.1 Local Level Optimisation: Problem Analysis

For the GDI engine calibration problem, the main objective was to minimise the fuel consumption over the NEDC drive cycle. However, considering that the new emission legislation for GDI engine is introduced based on Pn (i.e. Euro VI), minimising the Pn emission was also considered as an objective. Another reason for minimising the Pn response was to reduce the need for an after-treatment hardware [46]. Accordingly, the idea proposed was to define the local optimisation problem as a trade-off optimisation problem between Pn and FC at each minimap. The relation between FC and Pn responses has not been studied before, although it is generally expected to observe a trade-off relation between the fuel consumption and emissions.

Therefore, a multi-objective optimisation problem was formulated with the objectives of minimising both fuel consumption and Pn, in subject to the operating constraints at each minimap point, as illustrated in Equation 6.1.

Objectives:
Minimise $FC\ (X_i)$
Minimise $Pn\ (X_i)$

With respect to
X_i

Subject to
$EVC_i - IVO_i \leq E_i$
$h(X_i) < 0$
$X_i^{lb} \leq X_i \leq X_i^{ub}$

Equation 6.1

i indicates the calibration minimap point (speed / load operating point) given in Table 4.1, X_i specifies the vector of calibration parameters at minimap point *i*, $FC\ (X_i)$ is the fuel consumption at minimap point *i*, and $Pn\ (X_i)$ denotes the Pn value at minimap point *i*. Given that the multi-objective optimisation process was conducted for the GDI engine case study using both single injection and double injection strategies, the calibration variables constituting the X_i vector can be defined as following:

$X_i = (IVO_i, EVC_i, FRP_i, SOI_i)$ For single injection models

$X_i = (IVO_i, EVC_i, FRP_i, SOI1_i, Delay_i, Split_i)$ For double injection models

The local constraints in Equation 6.1 are:

- **Valve overlap** ($EVC_i - IVO_i$)**:** This constraint specifies the maximum allowable valve overlap (E_i), which were determined for some of the minimap points during the Screening DoE stage described in Chapter 5, to avoid operating conditions with poor combustion stability. The maximum allowable valve overlap at each minimap point is summarised in Table 6.1.

Maximum Overlap	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6
Calibration Case 1	10 deg	30 deg	NA	NA	NA	NA
Calibration Case 2	10 deg	30 deg	NA	30 deg	NA	NA

Table 6.1: Upper limits for the overlap constraint at each minimap point

- **Convex hull** (*h*)**:** Convex hull [227] was used as an additional design space constraint during the optimisation process. Convex hull or convex envelope defines the minimal convex which contains all the feasible test data points as the final design space. This constraint helps to avoid extrapolation around the unknown part of design space, i.e. where not enough test point is generated such as design space corners. The Matlab MBC toolbox was used to apply the star-shaped [171] boundary limits on the GDI engine MBC models fitted in Chapter 5. This type of convex envelope is constructed by interpolation of all data points on boundary to have a wider design space range through a more flexible design space constraint. This envelope behaves similar to the typical convex hull, while provides more opportunity to explore the design space during the optimisation stage. In Equation 6.1, *h* determines the distance between a point and the convex hull constraint. So, when $h(X_i) < 0$, it means that the calibration solution is inside the convex hull.

- **Boundary limits:** Linear constraints were used to define the boundary limits for each calibration parameter (see Table 5.3 and Table 5.6). In Equation

6.1, *lb* and *ub* indicate the lower and upper boundary limits for each variable, respectively.

6.2.1.2 Local Level Optimisation: Implementation and Results

To solve the multi-objective optimisation problem illustrated in Equation 6.1, the NSGA II optimisation algorithm, which is an evolutionary multi-objective optimisation algorithm [173], was utilised. Accordingly, an NSGA II optimisation algorithm implementation was downloaded as an optimisation toolbox available for Matlab programming environment [247]. The fitted response models in Chapter 5 were also exported from the MBC toolbox into the Matlab workspace using the 'data structure' format. Then, the exported MBC models and the local constraints illustrated in Equation 6.1 were loaded into the NSGA II graphical user interface (GUI) to run the algorithm for each of the minimap points.

Figure 6.1 illustrates an example of FC – Pn trade-off at 1250 (rpm) - 125 (Nm) operating point, i.e. for both single and double injection models. For these trade-offs, the NSGA II algorithm was run for 200 iterations while the 'Population Size' was 100, 'Crossover Rate' was 0.8, 'Mutation Rate' was 0.05, and 'Elite Size' was 5. By analysing this figure, it can be concluded that:

(i) There is an explicit trade-off between Pn and FC responses that can be exploited to deliver a better calibration for Pn. For instance, for the single injection plot, a significant reduction in Pn (over 30% reduction, by reducing Pn from 6×10^5 to 4×10^5) can be achieved with a very small penalty (around 0.5%) in fuel consumption.

(ii) The use of double injections appears to offer an important opportunity to reduce the fuel consumption (i.e. approximately 2%) in this minimap point, however; it does not deliver a better calibration solution in the sense to meet the forthcoming Pn emission legislation.

Similar trade-off trends were observed at the other engine speed / load operating points tested for the GDI engine case study. All the results were submitted to the CREO project as a report 'CREO 20082013: Preliminary Report on Optimisation Stage (Optimisation Plots)'.

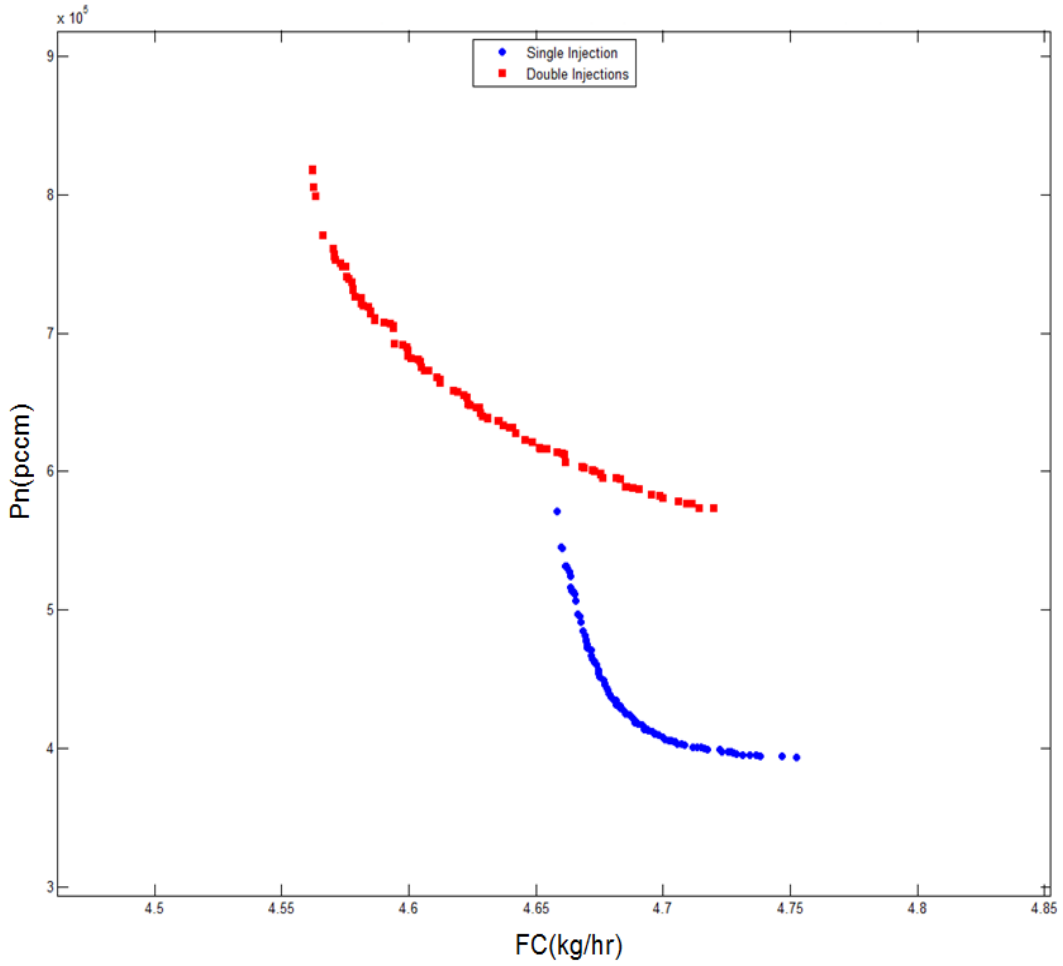


Figure 6.1: Illustration of Pn - FC trade-off study at 1250 (rpm) -125 (Nm)

In order to plot the location of NSGA II Pareto solutions within the design space, a Matlab code was written as an extension to the NSGA II optimisation toolbox, to map the Pareto solutions into 2-D coordinates for any possible combination of calibration parameters. In this program, a colour scheme was used to define which Pareto solutions are delivering a better Pn value (i.e. or worse FC value).

Figure 6.2 shows an example of mapping Pareto solutions within the possible 2-D coordinates, applied for the 1250 (rpm) - 125 (Nm) operating condition (i.e. Pareto frontier is shown in Figure 6.1). In this figure, blue crosses show the location of feasible DoE test points, the coloured points are indicating the Pareto solutions changing colour from black to white (i.e. a darker point indicates a lower Pn value / or higher FC value), and the lighter grey background illustrates the convex hull over the DoE test points within the boundary limits. Noteworthy, the convex hull was shown using the '*Convhulln.m*' function available in Matlab library.

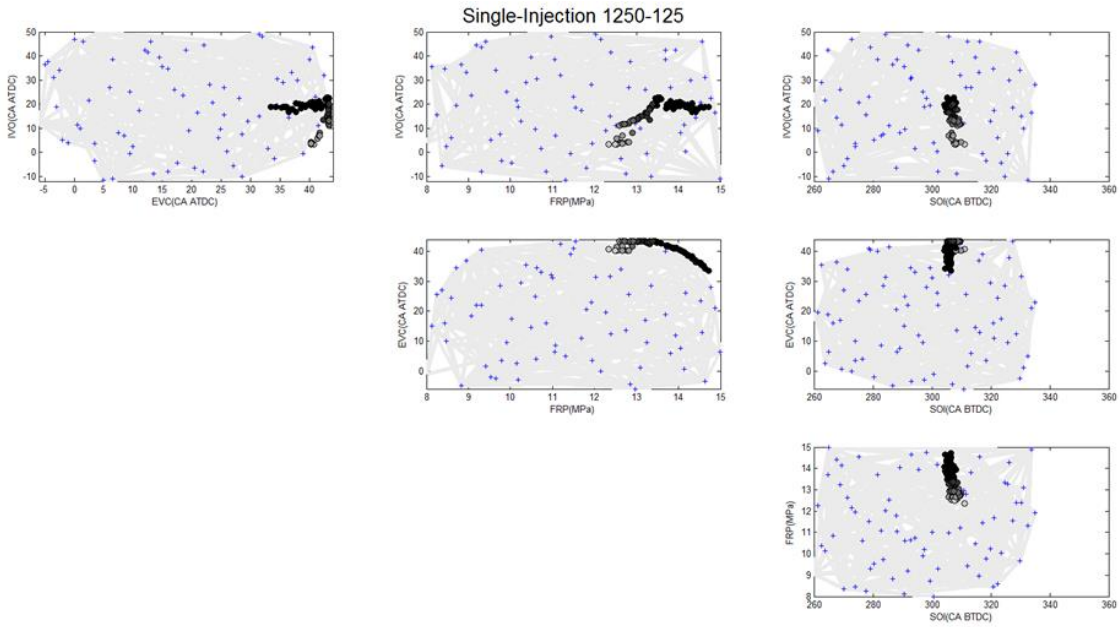


Figure 6.2: Mapping of Pareto solutions for the 1250 (rpm)-125 (Nm) minimap point with single-injection strategy within 2-D co-ordinates

Using the colour-map graphs, it was easier to compare the effect of each variable on the Pn and FC responses. For instance, in Figure 6.2 an increase in fuel rail pressure (FRP) has resulted in a decrease in Pn (due to better fuel atomization, as explained in Chapter 4) which are clearly monitored by darker points at higher FRP values.

Similar figures were plotted for all the minimap points, i.e. for both single injection and double injection strategies.

6.2.2 Global Level Optimisation

6.2.2.1 Selection of Local Solutions

After analysing the Pareto solutions within the design space for each minimap point, a number of calibration settings (i.e. 4 solutions) were selected on each of the Pareto frontiers as the potential local optimum solutions. These solutions were selected in a way to be representative of the trade-off between FC and Pn responses at each of the minimap points, which does not necessarily consider the spread of calibration solutions within the design space. As an example, Figure 6.3 illustrates the selected local candidates at 1250 (rpm) - 125 (Nm) minimap point with single injection strategy.

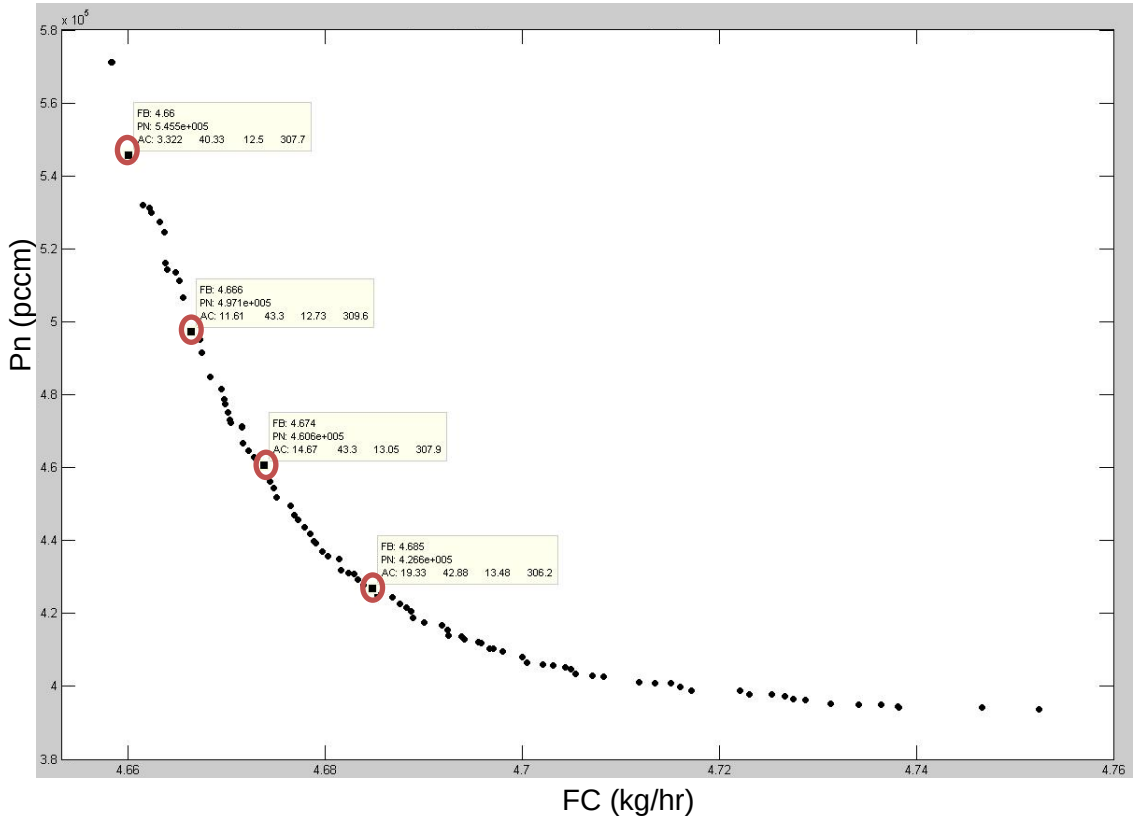


Figure 6.3: Local candidates for 1250 – 125 operating point with single injection

The selected candidate solutions on Figure 6.3 are also summarised in Table 6.2. Similar process was used for all the tested minimap points of the GDI engine case study (i.e. 6 minimap points using single injection strategy and 4 minimap points using double injections strategy). The local calibration candidates for each minimap point were summarised in separate tables, i.e. similar to Table 6.2, which are shown Appendix II.

Candidate	IVO (ATDC)	EVC (ATDC)	FRP (MPa)	SOI (BTDC)	FC (kg/hr)	Pn (pccm)
1	3.2	40.3	12.5	307	4.660	5.45E05
2	11.6	43.3	12.7	309	4.666	4.97E05
3	14.7	43.3	13.05	307	4.674	4.61E05
4	19.3	42.8	13.48	306	4.685	4.27E05

Table 6.2: Local candidates for 1250-125 operating point with single injection

6.2.2.2 Global Level Optimisation: Problem Analysis

In the global level, the main objective was to select the best combination of solutions among all the possible combinations of candidates, which could deliver the minimum overall fuel consumption and a low overall Pn value in comparison with the other combinations. The overall fuel consumption (FC_T) and overall Pn (Pn_T) was calculated using the following equations;

$$FC_T = \sum_{i=1}^{6} w_i \times FC\ (X_i)$$
$$Pn_T = \sum_{i=1}^{6} w_i \times Pn\ (X_i) \qquad \text{Equation 6.2}$$

Where w_i indicates the equivalent residency time for minimap *i* (given in Table 4.1) over the NEDC drive cycle. In this equation, X_i denotes the vector of calibration parameters at minimap *i*, and $FC\ (X_i)$ and $Pn\ (X_i)$ are the predicted fuel consumption and particulate number at minimap *i*.

The main calibration preference to be considered during the global optimisation was to deliver a smooth ECU map (i.e. calibration solutions), which is an engineering requirement since a non-smooth calibration solution can result in customer perceived transient driveability issues. A smooth calibration solution can be delivered by restricting large changes in the calibration parameters with a swift load increase. Accordingly for the GDI engine, the maximum actuator change related to transitions between each 2 minimap points *i* and *j* was restricted to a fraction (*k*) of the design space. The actuator change requirement for the GDI engine was mathematically formulated as following.

$$Max\left(\left|X_i - X_j\right|\right) \leq k \times \left|X_i^{max} - X_i^{min}\right| \qquad 0 < k < 1 \qquad \text{Equation 6.3}$$

Where, X_i^{max} and X_i^{max} are defining the minimum and maximum limits for the corresponding calibration parameter X_i, respectively.

Consequently, the global objective function corresponding to this multi-objective optimisation problem was formulated using the 'Weighted sum' method [141] (explained in Chapter 3), as illustrated in Equation 6.4.

Objective:

Minimise $a \times \sum_{i=1}^{6} w_i \times FC\ (X_i) + b \times \sum_{i=1}^{6} w_i \times Pn\ (X_i)$

With respect to

X_i

Subject to

$$g_0 = Max\left(\left|X_i - X_j\right|\right) - k \times \left|\mathrm{X}_i^{max} - \mathrm{X}_i^{min}\right| \leq 0$$

where i & $j = 1, \ldots, 6\ (i \neq j)$

$$X_i^{lb} \leq X_i \leq X_i^{ub}$$

Equation 6.4

In this equation, the global objective is formulated as the weighted sum of FC and Pn responses. The main reason of selecting the 'Weighted sum' optimisation method was the simplicity of implementing this multi-objective optimisation strategy, and also the capability of controlling the relation between the objectives using a weight vector $[a\ b]$. Considering that the Pn and FC responses are having different units, both responses were standardised between -1 and 1 [-1 1], and the focus of optimisation algorithm on each of the responses was controlled by changing '***a***' and '***b***' coefficients. In this research, '***a***' and '***b***' were set as: $a = 4$ and $b = 1$, which means the final calibration solution emphasises on minimising the FC response 4 times more than minimising Pn, due to calibration requirements for the GDI engine case study (CREO project) to pressurise improvement in overall fuel consumption. Moreover, in this equation *k* was set to 0.5, which in effect results in constraining the maximum actuator change related to any transition between any 2 minimap points *i* and *j* to half of the design space, i.e. this strategy is similar to the one adopted by Singh et al [2].

After formulating the global objective function, the next step was to evaluate the objective function for all the possible combinations of local solutions. Given that 4 candidates were selected for each of the 6 minimap points, there were (4^6 = 4096) possible combinations to be evaluated. Subsequently, the global solution with the best fitness value was selected subject to the drivability constraint (g_0), shown in Equation 6.4.

It should be noted that as explained in Chapter 4, two calibration cases were considered for the GDI engine case study (see Table 4.4):

- 'Case 1', for which the global optimisation was over all the 6 minimap points tested using single injection strategy.
- 'Case 2', for which the global optimisation was over all the minimap points tested using the double injection strategy (i.e. minimaps 3 to 6), and the minimap points where double injection was not accessible (i.e. minimap points 1 and 2).

In this chapter, similar optimisation strategies were conducted for both calibration cases, however; the main difference was the number of calibration parameters for each case, i.e. calibration 'Case 1' has 24 linking variables, and calibration 'Case 2' has 32 linking variables.

6.2.2.3 Global Level Optimisation: Implementation

To define the optimum global solution, it was required to exhaustively consider all the possible combinations of the local candidates. A Matlab code was developed to evaluate the objective function (i.e. fitness function), illustrated in Equation 6.4, for each of the possible combinations. In this Matlab code, in order to avoid selecting a solution that violates the drivability constraint (g_0), which was set to half of the design space (k = 0.5), the objective function was formulated as:

$$fitness = a \times \sum_{i=1}^{6} w_i \times FC\,(X_i) + b \times \sum_{i=1}^{6} w_i \times Pn\,(X_i) + \emptyset$$

where

$$\emptyset = \begin{cases} 0 & if\ Max\left(\left|X_i - X_j\right|\right) - k \times \left|\mathrm{X}_i^{max} - \mathrm{X}_i^{min}\right| \leq 0 \\ 1000 & if\ Max\left(\left|X_i - X_j\right|\right) - k \times \left|\mathrm{X}_i^{max} - \mathrm{X}_i^{min}\right| > 0 \end{cases}$$

$$i\ \&\ j = 1, \ldots, 6\ (i \neq j)$$

Equation 6.5

In Equation 6.5, the objective function given in Equation 6.4 was modified using a penalty function (∅), which adds a large number to the fitness function if the possible global solution cannot satisfy the drivability constraint. In this equation, ∅ was set as 1000, after a number of try and error, to detect when the drivability constraint is violated. In the developed Matlab code, the response values were

evaluated using the exported response models from the MBC toolbox into the Matlab workspace.

Therefore, using Equation 6.5 as the fitness function, the feasible solution (i.e. combination of local candidates) with the least fitness value was selected as the optimum global solution.

6.2.2.4 Global Level Optimisation: Results

I. Calibration Case 1

The global optimisation solution for the calibration 'Case 1' is summarised in Table 6.3. This table represents the optimum setting for each of the 24 calibration parameters (i.e. IVO, EVC, FRP and SOI for each of 6 minimap points), and the corresponding calibration objectives, i.e. FC (kg/hr) and Pn (pccm), for each engine speed / load operating condition. The total FC and Pn responses over the reference drive cycle (i.e. NEDC drive cycle) is also shown in this table.

Speed / Load (rpm – Nm)	**IVO** (ATDC)	**EVC** (ATDC)	**FRP** (MPa)	**SOI** (BTDC)	**FC** (kg/hr)	**Pn** (pccm)
700-28	35.2	14.8	12.3	294	1.374	5.24E+2
1500-41	34.7	23.6	11.9	314	3.212	3.58E+2
1250-125	11.6	43.3	12.7	309	4.666	4.97E+5
1500-105	29.6	37.0	14.1	318	5.154	6.58E+5
2000-81	25.4	38.45	14.7	306	5.848	5.67E+5
2000-199	37.0	41.1	12.4	288	10.693	1.61E+6
				Overall	2.738	1.36E+5

Table 6.3: 2-stage calibration optimisation solution for the calibration Case 1

Figure 6.4 illustrates the 2-stage calibration solutions as a parallel axis plot, comparatively with the baseline calibration setting (see Table 4.5) at each minimap point, for all calibration variables. This figure shows that the 2-stage framework delivered a smooth actuator map for all the calibration variables, with significant improvements for the EVC and FRP. As an example, for the 2-stage solution, the maximum change in FRP along all the minimap points is 2.9 MPa, however; the baseline setting for FRP shows a significant change from 3 to 14

MPa, during the possible transfer from minimap 1 to minimap 6, which is not desirable due to drivability issues.

Figure 6.5 and Figure 6.6 compare the 2-stage optimisation solution versus the baseline calibration setting, in terms of percentage improvement in overall FC and Pn values. The improvements over the drive cycle show that the 2-stage approach outperforms the baseline calibration setting for both overall FC and Pn responses, which were significantly improved by 6.82 % and 74.79%, respectively.

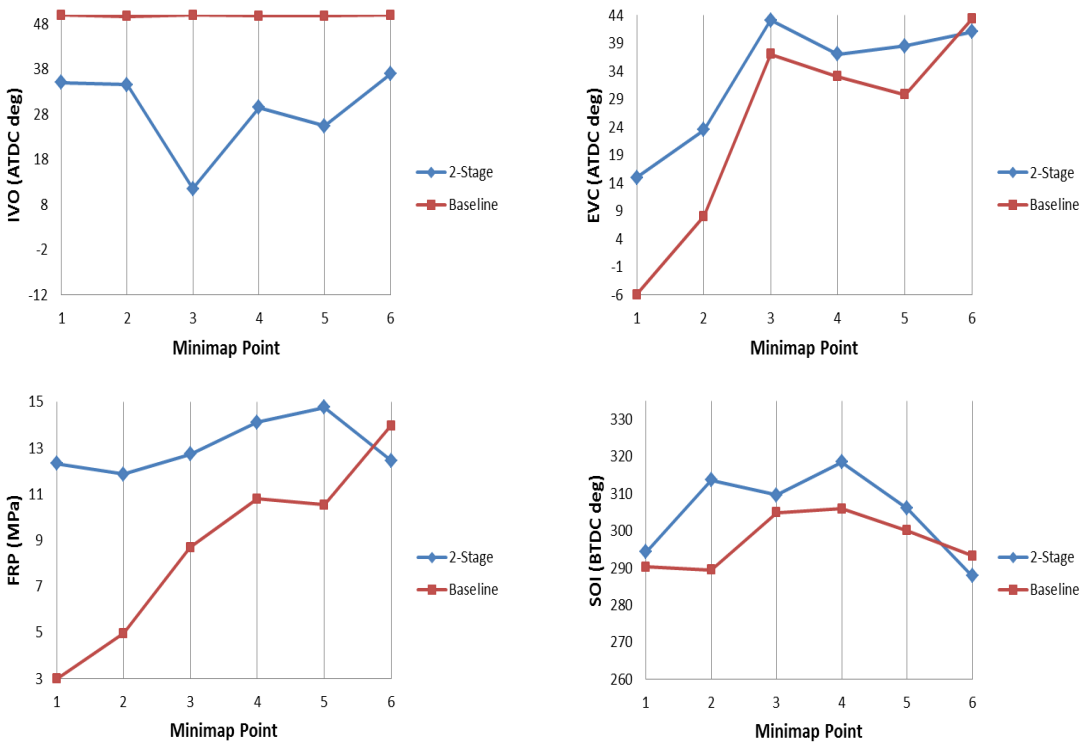


Figure 6.4: Actuator setting at each minimap for baseline and 2-stage calibration settings

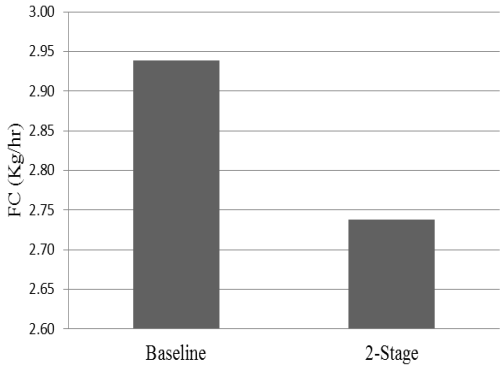


Figure 6.5: Comparing total FC over NEDC drive cycle

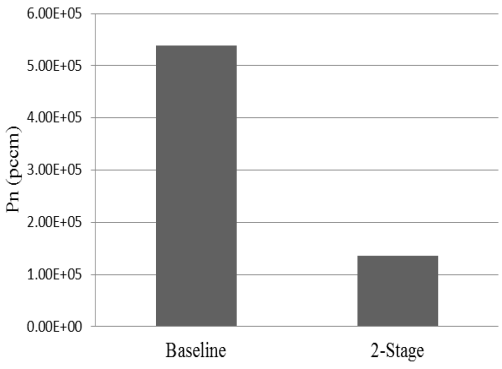


Figure 6.6: Comparing total Pn over NEDC drive cycle

II. Calibration Case 2

The same analysis of results was conducted for the calibration Case 2. The global optimum solution for each calibration parameter is illustrated in Table 6.4. Also, the corresponding FC (kg/hr) and Pn (pccm) to each minimap solution, and the total FC and Pn over the NEDC drive cycle is illustrated in this table.

Minimap (rpm – Nm)	**IVO** (ATDC)	**EVC** (ATDC)	**FRP** (MPa)	**SOI1** (BTDC)	**Delay** (ms)	**Split** (%)	**FC** (kg/hr)	**Pn** (pccm)
700-28	35.2	14.8	12.3	294	NA	NA	1.374	5.24E+2
1500-41	30.5	24.6	11.1	311	NA	NA	3.192	6.59E+3
1250-125	12.5	39.4	14.0	337	3.07	0.54	4.668	6.03E+5
1500-105	6.4	34.5	12.9	345	7.03	0.57	5.036	1.11E+6
2000-81	23.2	38.3	13.1	347	1.57	0.54	5.96	1.19E+6
2000-199	23.1	36.9	13.4	354	2.28	0.56	10.727	1.64E+6
						Overall	2.729	1.97E+5

Table 6.4: 2-stage calibration optimisation solution for the calibration Case 2

Figure 6.7 illustrates the 2-stage calibration solutions for the calibration Case 2 as a parallel axis plot, comparatively with the baseline calibration setting (see Table 4.5). Since the baseline solution is based on single injection strategy, Figure 6.7 only illustrates IVO, EVC, FRP and SOI variables for the comparison. This figure shows that the 2-stage framework delivered a smooth actuator map for all the calibration variables, i.e. particularly for the EVC and FRP (similar to calibration Case 1).

Figure 6.8 and Figure 6.9 compare the 2-stage optimisation solution for calibration Case 2 versus the baseline calibration setting, in terms of percentage improvement in overall FC and Pn values. These figures show that the 2-stage optimisation solution outperforms the baseline calibration setting over both overall FC and Pn responses, which were improved by 7.13 % and 63.35%, respectively.

Comparing the solutions obtained from each of the calibration cases (i.e. ‘Case 1’ versus ‘Case 2’) it was observed that using double injection strategy reduced the overall fuel consumption marginally, however; the overall Pn was increased significantly. In other words, using double injection strategy for the GDI engine does not appear to improve the Pn emissions.

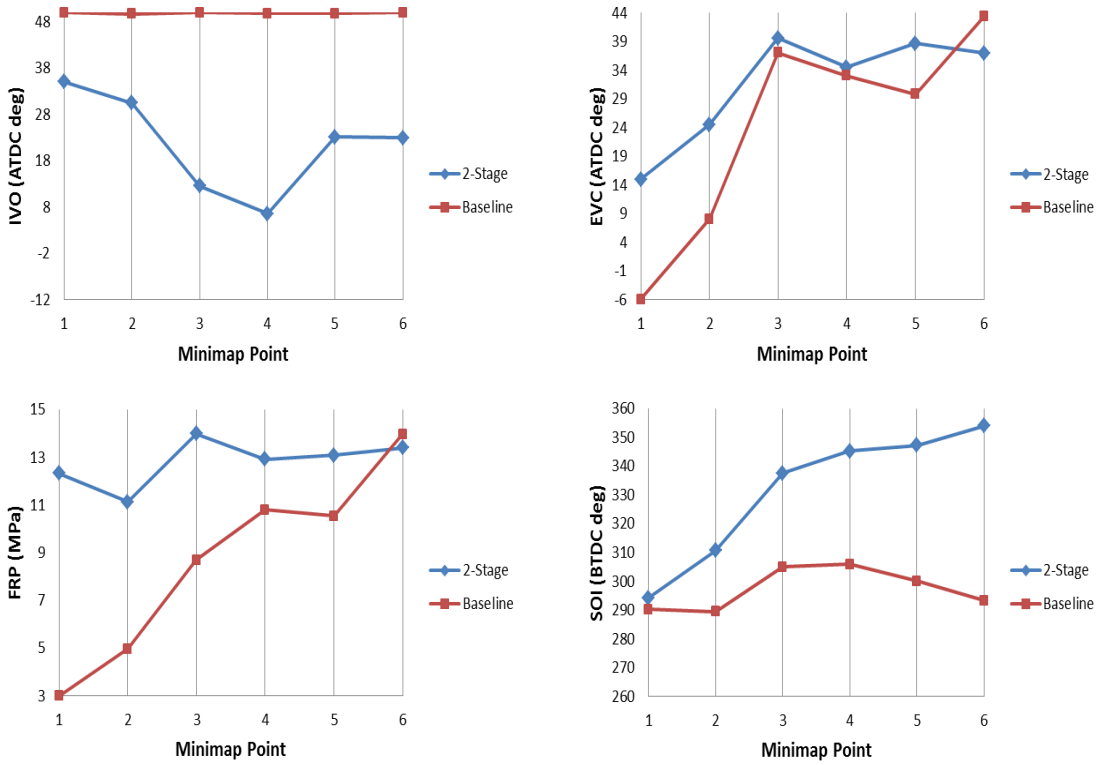


Figure 6.7: Actuator setting at each minimap for baseline and 2-stage calibration settings

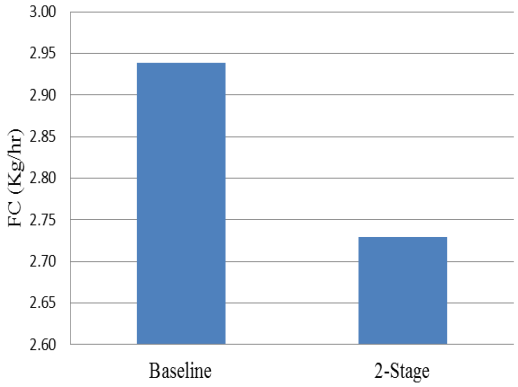


Figure 6.8: Comparing total FC over NEDC drive cycle between 2-stage approach and baseline setting for calibration Case 2

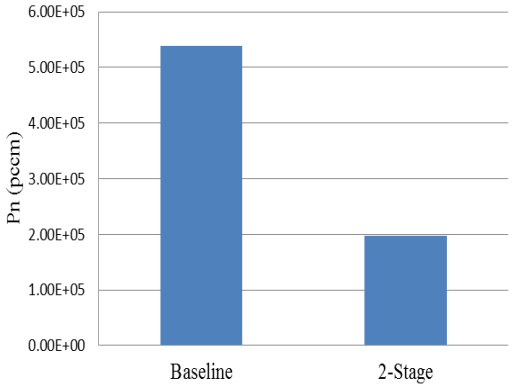


Figure 6.9: Comparing total Pn over NEDC drive cycle between 2-stage approach and baseline setting for calibration Case 2

6.2.3 Discussion of 2-Stage Optimisation Approach

Although the 2-stage optimisation strategy was successfully implemented for the GDI engine case study, it has several shortcomings that are required to be addressed to improve the calibration optimisation process. Some of these shortcomings are outlined as following:

- Finding the global calibration solution is generally a very time consuming and iterative process. As an example, for the GDI engine problem that 4 solutions were carried over to the global level from each of the 6 minimap points, there were 4^6 possible solutions that required to be evaluated to select the best solution. So, in a case that none of the combinations could deliver a feasible solution this process had to be repeated again from the local optimisation level to select more solution candidates.

- The global optimisation process is not goal focused, in a sense that instead of minimising the main objectives this process is designed to find the best combination of the local solutions. So, the global level optimisation for the GDI engine was limited to several possible calibration solutions not necessarily the true global solution.

- Selecting the best global solution requires prior knowledge regarding the engine performance (i.e. calibrator's experience).

- Considering that the complexity of the calibration optimisation problem is increasing by introducing new equipment to the engine (i.e. the number of calibration parameters and also the calibration requirements are increasing), providing the local optimisation candidates from the Pareto trade-offs and subsequently delivering the global optimum solution is getting more complex and far more challenging.

6.3 Application of Multidisciplinary Design Optimisation Frameworks for the GDI Engine Steady State Calibration Optimisation Problem

Considering the shortcomings of the conventional 2-stage optimisation strategy, underlines the need for a better optimisation method to solve the GDI engine calibration optimisation problem. In order to enhance the optimisation strategy, the two levels of optimisation problem ('Local' and 'Global' levels) should ideally be approached and solved concurrently, such that both global (i.e. over the drive cycle) and local (i.e. at each minimap point) benefits are achieved.

Accordingly, as discussed in Chapter 4, three multidisciplinary design optimisation strategies (i.e. MDF, CO, and ATC) were applied to formulate the

GDI engine calibration optimisation problem, which are coherent with the 2-level structure of the steady state calibration optimisation problems.

6.3.1 Multidisciplinary Feasible Framework (MDF)

6.3.1.1 MDF Approach: Problem Analysis

MDF is the simplest multidisciplinary design optimisation method which is derived from the All-At-Once (AAO) optimisation strategy. This framework uses a system optimiser to control all the design variables, objective functions and design constraints [39].

The GDI engine calibration problem was associated with the MDF framework by formulating the engine performance over all 6 minimap points (i.e. drive cycle) as the *system level optimisation*, while treating each minimap point as a *subsystem analyser*. A subsystem analyser is responsible to define the response functions at each minimap point for the linking variables $\boldsymbol{y_{si}}$ transferred from the system level optimiser. Therefore, the nomenclature of the non-hierarchical formulation of the GDI engine calibration optimisation process using MDF is detailed in Table 6.5.

Level	Decomposition of GDI engine calibration optimisation problem
System level Optimisation (*over a drive cycle*)	*Objectives:* Minimise Overall fuel consumption (FC_T) Minimise Overall particulate number (Pn_T) *Constraints:* Drivability constraint Overlap constraint at each minimap point Convex hull constraint at each minimap point Linear constraints *Design variables:* Calibration parameters over 6 minimap points

Table 6.5: GDI engine calibration optimisation problem using MDF

Accordingly, the organisation of the GDI engine calibration optimisation problem using the MDF optimisation framework is illustrated in Figure 6.10. In this figure, the system level optimisation is shown by the rectangle, where the rounded rectangle represents the subsystem analyser *i*. Within the MDF approach, the system optimiser optimises the system level objective function (i.e. consists of

both fuel consumption and Pn responses) in subject to the system level constraints (i.e. drivability constraint over all the minimap points, overlap and convex hull constraints at each minimap point). In order to evaluate the system level objective function, the system level optimiser transfers the vector of linking variables y_{si} to each subsystem analyser *i*. Then, the analyser uses the transferred linking parameters to evaluate the FC and Pn response models, which were developed in Chapter 5, and send the response values back to the system level to evaluate the system level objective function. This iterative process continues until the system level optimiser (i.e. optimisation algorithm) terminates.

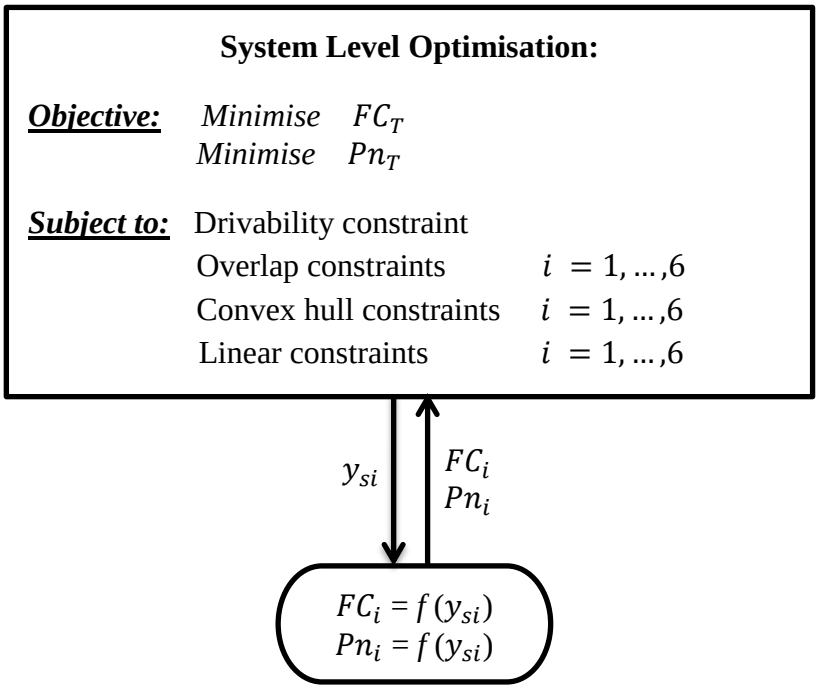


Figure 6.10: MDF Optimisation process for the GDI engine calibration optimisation problem

Equation 6.6 illustrates the MDF formulation for the GDI engine calibration problem. The 'Weighted sum' multi-objective optimisation strategy [141] was used to formulate the objective function, aiming to minimise both fuel consumption and Pn.

Objective:

Minimise $f_0 = a \times \sum_{i=1}^{6} w_i \times FC\,(y_{si}) + b \times \sum_{i=1}^{6} w_i \times Pn\,(y_{si})$

With respect to

y_{si}

Subject to:

$$g_0 = Max\left(\left|y_{si} - y_{sj}\right|\right) - k \times \left|\mathrm{y}_{si}^{max} - \mathrm{y}_{si}^{min}\right| \leq 0$$

where i & $j = 1, \ldots, 6\ (i \neq j)$

$$g1_i = EVC(y_{si}) - IVO(y_{si}) \leq E_i \qquad i = 1, \ldots, 6$$

$$g2_i = h(y_{si}) \leq 0 \qquad i = 1, \ldots, 6$$

$$y_{si}^{min} \leq y_{si} \leq y_{si}^{max} \qquad i = 1, \ldots, 6$$

Equation 6.6

f_0 is the system level objective function, w_i indicates the equivalent residency time for minimap *i* (given in Table 4.1), y_{si} denotes the vector of linking calibration parameters at minimap *i*, $FC\,(y_{si})$ and $Pn\,(y_{si})$ are the predicted fuel consumption and particulate number at subsystem analyser *i*, g_0 is the drivability constraint, $g1_i$ defines the overlap constraint at minimap *i*, $g2_i$ indicates the convex hull constraint at minimap *i*, and y_{si}^{min} and y_{si}^{max} are the lower and upper boundary limits for the linking variables. In this equation, E_i is the maximum allowable valve overlap, which was determined for some of the minimap points, summarised in Table 6.1. Also, $h(y_{si}) \leq 0$ ensures that the calibration solution is within the convex hull envelope.

Similar to the 2-stage optimisation approach, illustrated in Equation 6.4, the objective function coefficients in Equation 6.6 were set as: $a = 4$ and $b = 1$. Also, *k* was set to 0.5, to constraint the maximum allowable actuator change related to any transition between any 2 minimap points *i* and *j* to half of the design space.

Worth mentioning, y_{si} includes 4 variables for the single injection models $(\mathrm{IVO_i\ EVC_i\ FRP_i\ SOI_i})$, and 6 variables $(\mathrm{IVO_i\ EVC_i\ FRP_i\ SOI1_i\ Delay_i\ Split_i})$ for the double injection models. Accordingly, the system optimiser was responsible to optimise 24 calibration parameters for the calibration 'Case 1', and 32 parameters for the calibration 'Case 2'.

6.3.1.2 MDF Approach: Implementation

The MDF optimisation framework for the GDI engine case study was implemented in MATLAB® programming environment.

For implementation, the required information regarding the GDI engine case study was loaded into the program 'Main Body', including:

- Upper and lower limits for the FC and Pn responses using a matrix. This information was gained during the data collection process in Chapter 5.
- Speed and load operating conditions for each minimap point using a matrix.
- The residency weight for each minimap, as shown in Table 4.1, using a matrix.
- Engine response models developed in Chapter 5, which were exported from the Matlab MBC toolbox into the Matlab workspace using the Matlab 'data structure' format.
- Linear constraints (i.e. boundary limits) for each of the calibration parameters using the Matlab 'Cell' format.

In order to explain how the program was developed, the components of MDF framework and the interconnections among them are visualised using the extended design structure matrix (XDSM) [231] diagram, which was explained previously in Chapter 5 in conjunction with the PermGA algorithm. The XDSM diagram for the MDF approach is illustrated in Figure 6.11.

For the system level optimisation, at **level 0**, system optimiser starts the searching process from an initial point which is transferred to the optimiser as an input $y^{(0)}$. In this implementation, either gradient based or global optimisation algorithms could be employed as the system optimiser. However, given that all the response functions are explicitly available, and the engineering preference for computation speed / fast convergence, a gradient based sequential quadratic programming (SQP) algorithm, based on the *fmincon* Matlab function [248], was employed to solve the MDF problem at the system level.

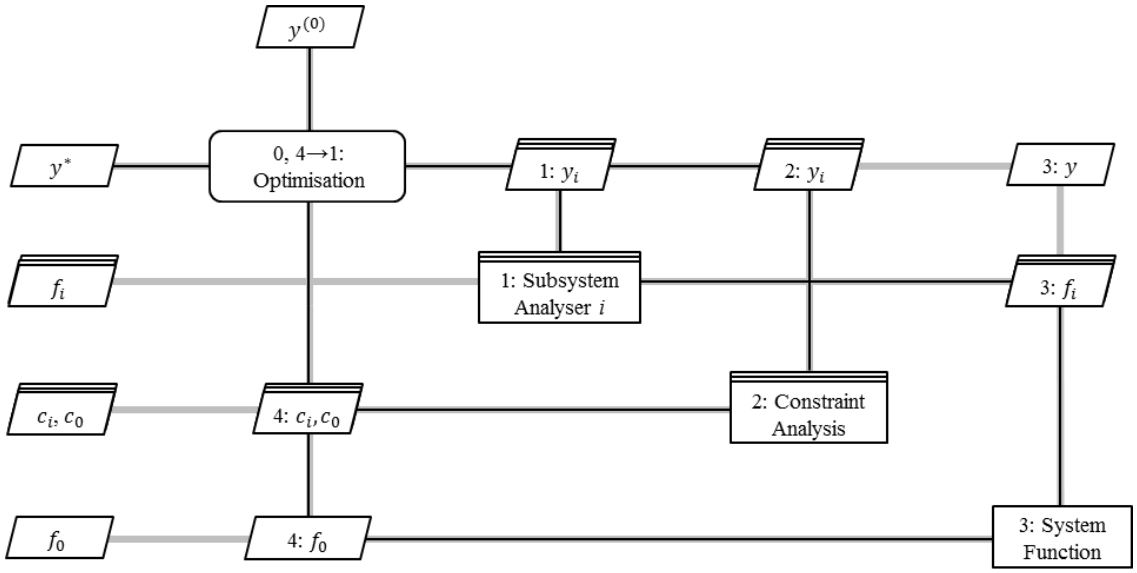


Figure 6.11: Illustration of MDF strategy for solving the steady state engine calibration optimisation problems using the XDSM graph

The optimisation algorithm transfers the optimum solution for the calibration parameters into levels 1 and 2. At **level 1**, the subsystem analysers use the transferred data y_i to evaluate the response values f_i (i.e. Pn and FC) for each minimap point *i*. These values are then transferred into **level 3** to evaluate the system level function f_0, as mathematically shown in Equation 6.6. The constraints at each minimap point and over all the minimap points (see Equation 6.6) are analysed at **level 2**, in parallel with level 1. In Figure 6.11, *c* is the vector of constraints, including both inequality and equality constrains. Thus, for the MDF formulation, c_i represents the constraints at each minimap point *i* (i.e. $g1_i$ and $g2_i$), and c_0 represents the constraint over all the minimap points (i.e. g_0). Afterwards, the outputs from level 2 (i.e. constraint values) and level 3 (i.e. system level objective function) are transferred into **level 4**, or 'Main Body, to check whether the solution satisfies the convergence requirements set for the *fmincon* algorithm. If the solution meets the requirements, the optimisation algorithm terminates and the final solution y^* is the optimum calibration solution. Otherwise, the optimisation algorithm uses the transferred solution y^* as the initial search point for another iteration of the optimisation process.

The pseudocode for the developed MDF framework, illustrated in Figure 6.11, is listed in Table 6.6.

Input: Initial design variables ($y^{(0)}$)

Output: Optimal variables (y^*), objective function ($f_0{}^*$), constraint values (c^*)

0: Initiate the system level optimisation iteration

Repeat

for each subsystem (discipline) *i* do

1: Evaluate objective function (f_i) using subsystem analyser *i*

end for

for each subsystem (discipline) *i* do

2: Evaluate constraint (c_i)

end for

2: Evaluate the system level constraint (c_0)

3: Evaluate the system level function (f_0) based on the evaluated subsystem objective functions (f_i)

4: Send the system level objective function (f_0) and constraint analysis values (c_0, c_i) into the system optimiser to check the algorithm convergence

Until 4→1: Optimisation has converged

Table 6.6: MDF Pseudo Code for Engine Applications

6.3.1.3 MDF Approach: Results

I. Calibration Case 1

Considering that the performance of gradient based algorithms, such as *fmincon*, can be affected by the location of algorithm initial search point, the MDF optimisation was run with 5 random initial solutions and the best solution was considered as the optimum solution. The MDF framework converged to the optimum solution after 5 iterations in 179 seconds. The optimisation results for the calibration Case 1 using the MDF architecture are illustrated in Table 6.7. This table summarises the optimum setting for each of the 24 calibration parameters, and the corresponding calibration objectives, i.e. FC (kg/hr) and Pn (pccm), for each engine speed / load operating condition. The total FC and Pn responses over the NEDC drive cycle is also shown in this table.

Speed / Load (rpm – Nm)	**IVO** (ATDC)	**EVC** (ATDC)	**FRP** (MPa)	**SOI** (BTDC)	**FC** (kg/hr)	**Pn** (pccm)
700-28	32.0	15.0	12.3	294.5	1.376	4.52E+2
1500-41	31.5	11.5	13.1	296.1	3.205	6.18E+3
1250-125	2.0	36.5	13.5	306.5	4.680	4.72E+5
1500-105	2.0	36.5	10.5	278.3	5.072	1.22E+6
2000-81	2.1	36.5	13.5	299.3	5.860	8.34E+5
2000-199	2.0	36.4	10.5	303.4	10.741	2.89E+6
				Overall	2.734	2.06E+5

Table 6.7: MDF optimisation solution for the GDI engine calibration case 1

Figure 6.12 illustrates both the MDF solution and the baseline setting (see Table 4.5) as a parallel axis plot at each minimap point, for all the calibration variables. Considering the maximum change in the calibration parameters associated to the transitions between any 2 minimap points, it is apparent that the MDF optimisation solution delivered a ‘smoother’ calibration.

Figure 6.13 and Figure 6.14 compare the MDF optimisation solution versus the baseline calibration setting, in terms of percentage improvement in the overall fuel consumption and Pn, respectively. Accordingly, it can be seen that MDF approach reduced the overall fuel consumption by 6.95 %, and overall Pn by 61.72 %, compared to the baseline calibration setting.

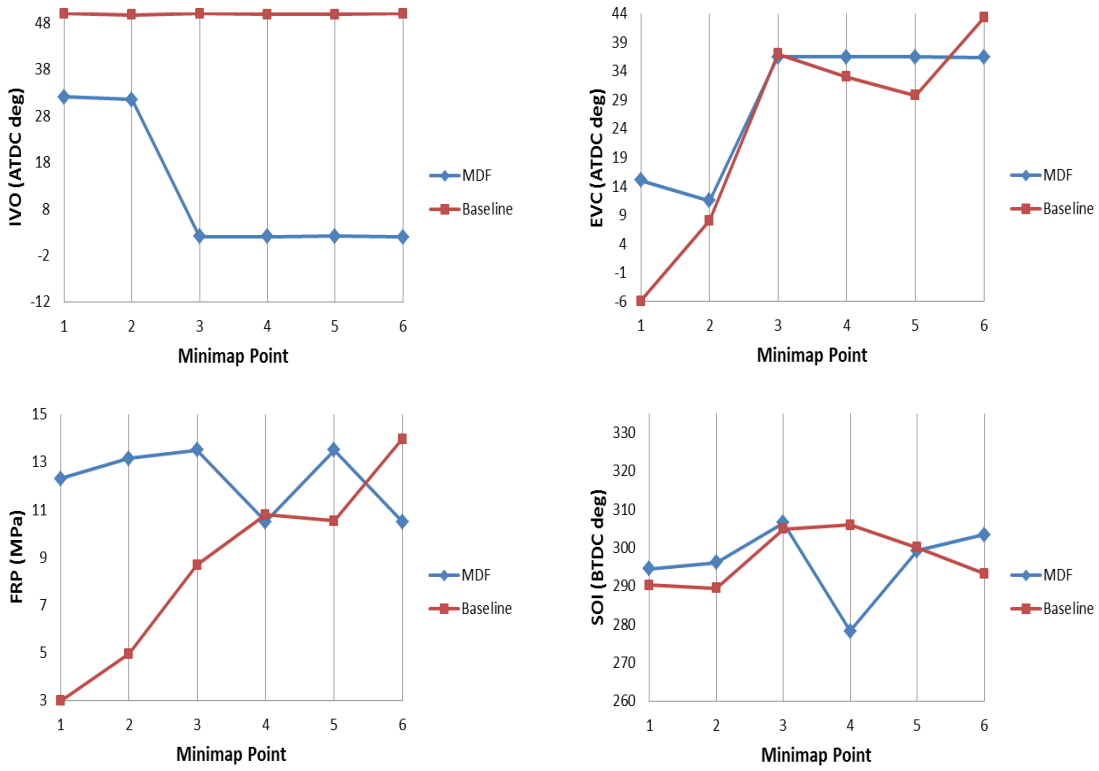


Figure 6.12: Actuator setting at each minimap for both baseline and MDF approaches

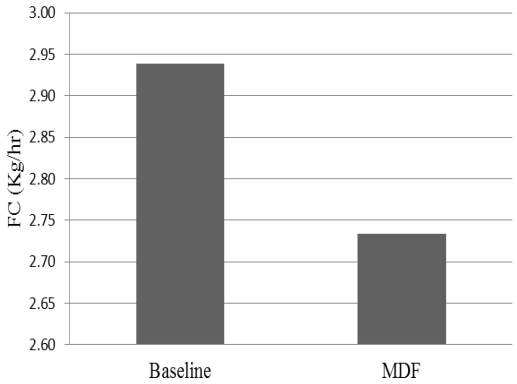


Figure 6.13: Comparing total FC over NEDC drive cycle

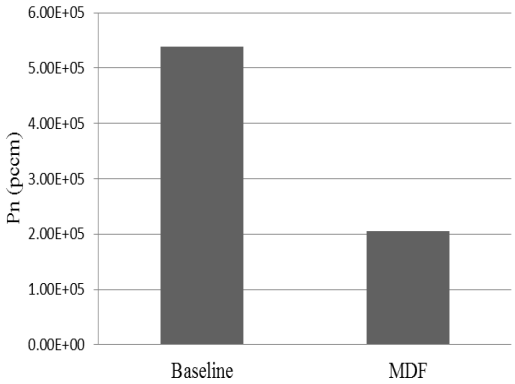


Figure 6.14: Comparing total Pn over NEDC drive cycle

II. Calibration Case 2

Similar to calibration Case 1, the MDF optimisation was run with 5 random initial solutions. For this case, the MDF framework converged to the optimum solution after 6 iterations in 216 seconds. The MDF optimisation solution for the GDI engine calibration Case 2 is illustrated in Table 6.8. The corresponding FC (kg/hr) and Pn (pccm) responses to each minimap solution, and the total FC and Pn over the NEDC drive cycle is also summarised in this table.

Minimap (rpm – Nm)	**IVO** (ATDC)	**EVC** (ATDC)	**FRP** (MPa)	**SOI1** (BTDC)	**Delay** (ms)	**Split** (%)	**FC** (kg/hr)	**Pn** (pccm)
700-28	41.0	19.0	12.0	298.0	NA	NA	1.374	1.88E+3
1500-41	32.5	11.5	13.1	295.7	NA	NA	3.205	6.03E+3
1250-125	15.0	36.5	14.0	317.1	4.6	0.33	4.693	7.94E+5
1500-105	45.0	35.0	13.0	297.0	1.5	0.62	5.086	9.11E+5
2000-81	44.3	36.0	12.5	289.0	1.5	0.62	5.908	1.76E+6
2000-199	15.4	33.4	13.0	316.4	1.5	0.37	10.733	2.35E+6
						Overall	2.736	2.34E+5

Table 6.8: MDF optimisation solution for the GDI engine calibration case 2

Figure 6.15 illustrates the calibration parameters (i.e. IVO, EVC, FRP and SOI) for both the MDF solution and the baseline setting at each minimap point using a parallel axis plot. This figure shows that the MDF framework could deliver a smooth engine map for all the calibration variables.

Furthermore, the MDF solution was compared against the calibration setting in terms of percentage improvement in the overall FC and Pn objectives, as shown in Figure 6.8 and Figure 6.9. These figures show that the MDF approach outperforms the baseline calibration setting, improving the overall FC by 6.87% and overall Pn by 56.41%.

Comparing the MDF optimisation solutions achieved for the calibration Case 1 versus Case 2, illustrated in Table 6.7 and Table 6.8, it was observed that the double injection strategy could not enhance the GDI engine performance in neither of the key calibration objectives (i.e. overall FC and Pn).

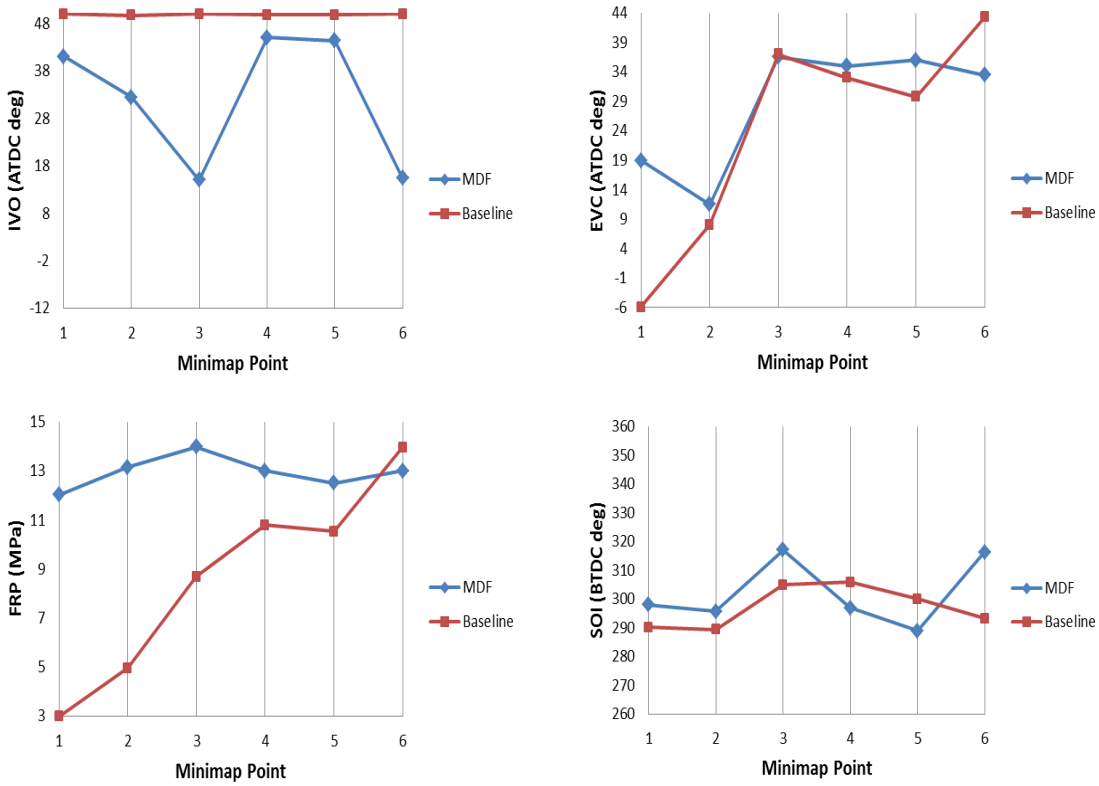


Figure 6.15: Actuator setting at each minimap for both baseline and MDF approaches

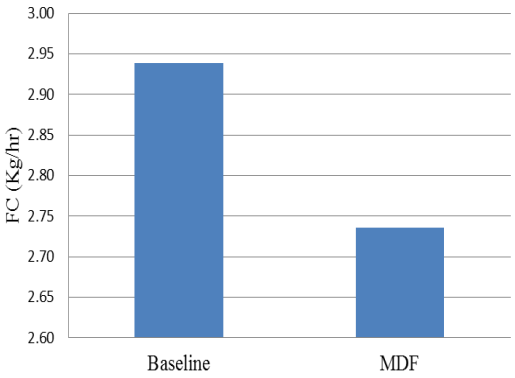


Figure 6.16: Comparing total FC over NEDC drive cycle between MDF approach and baseline setting for calibration case 2

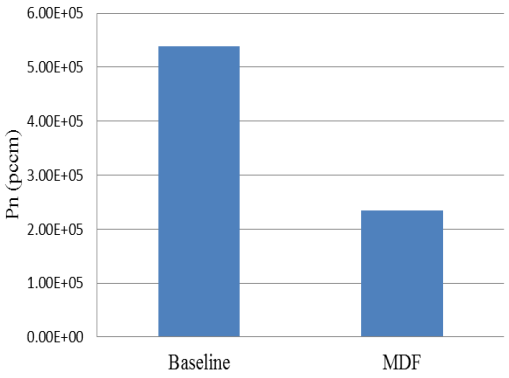


Figure 6.17: Comparing total Pn over NEDC drive cycle between MDF approach and baseline setting for calibration case 2

6.3.2 Collaborative Optimisation Framework (CO)

6.3.2.1 CO Approach: Problem Analysis

The CO optimisation architecture [32], [33], which is a bi-level hierarchal optimisation structure (i.e. a system level, and a subsystem level including a number of disciplines), was applied to formulate and solve the GDI engine steady state calibration optimisation problem. In this optimisation framework, each optimisation level has its own design variables, objective function and design constraints. The system level optimiser is responsible to minimise the design objective function, while the subsystem optimiser (i.e. at each discipline) has to minimise the discrepancy between the targets transferred from the system level and each discipline solutions (i.e. system inconsistency). In CO approach, the consistency between two levels of optimisation is maintained using an equality constraint at the system level.

Considering the 2-level structure of steady state engine calibration optimisation problem, a CO framework was associated with the engine calibration problem by treating each minimap point as a component of the *subsystem level optimisation* or discipline, and the engine performance over all minimaps (i.e. drive cycle) as the *system level optimisation* problem. Therefore, for the GDI engine case study, the calibration problem was decomposed into a bi-level CO framework, consisting of a system level and a subsystem level with 6 disciplines. The nomenclature of the GDI engine problem decomposition is detailed in Table 6.9.

Figure 6.18 illustrates the organisation of the applied CO framework for the GDI engine calibration optimisation problem. In this figure the rectangles show the optimisation levels, and the rounded rectangle represents the analysers which are responsible to define the response functions for the corresponding minimap point. Throughout the CO optimisation process, the system level optimises the system level objective function (i.e. consists of both fuel consumption and Pn responses) in subject to the system level constraints (i.e. drivability constraint over all the minimap points, and the consistency constraint). In the system level, the objective function is evaluated by sending the vector of linking variables y_{si} into each analyser *i*, to evaluate the response values for fuel consumption and

Pn using the response models developed in Chapter 5. Afterwards, the system level sends the optimisation output for each calibration parameter y_{ssi}^{U} to the corresponding discipline *i*. In the subsystem level at each discipline, the optimiser minimises the objective function, which is the discrepancy between the transferred calibration parameters from the system level y_{ssi}^{U} and the discipline calibration parameters y_{ssi}, in subject to the discipline constraints (i.e. overlap and convex hull constraints). Subsequently, the subsystem level sends back the discipline solutions y_{ssi}^{L} into the system level. This iterative process continues until the system level optimiser (i.e. optimisation algorithm) terminates.

Level	Decomposition of GDI engine calibration optimisation problem
System level *over a drive cycle*	*Objectives:* Minimise Overall fuel consumption (FC_T) Minimise Overall particulate number (Pn_T) *Constraints:* Drivability constraint Consistency constraint Linear constraints *Design variables:* Calibration parameters over 6 minimap points
Interface	*Linking variables between system level and subsystem level i* Calibration parameters at subsystem level *i*
Subsystem level Discipline *i* *at each minimap point i*	*Objective:* Minimise the discrepancy to the cascaded target *Constraints:* Overlap constraint Convex hull constraint Linear constraints *Design variables:* Calibration parameters at minimap *i*

Table 6.9: Decomposition of GDI engine calibration optimisation problem using CO architecture

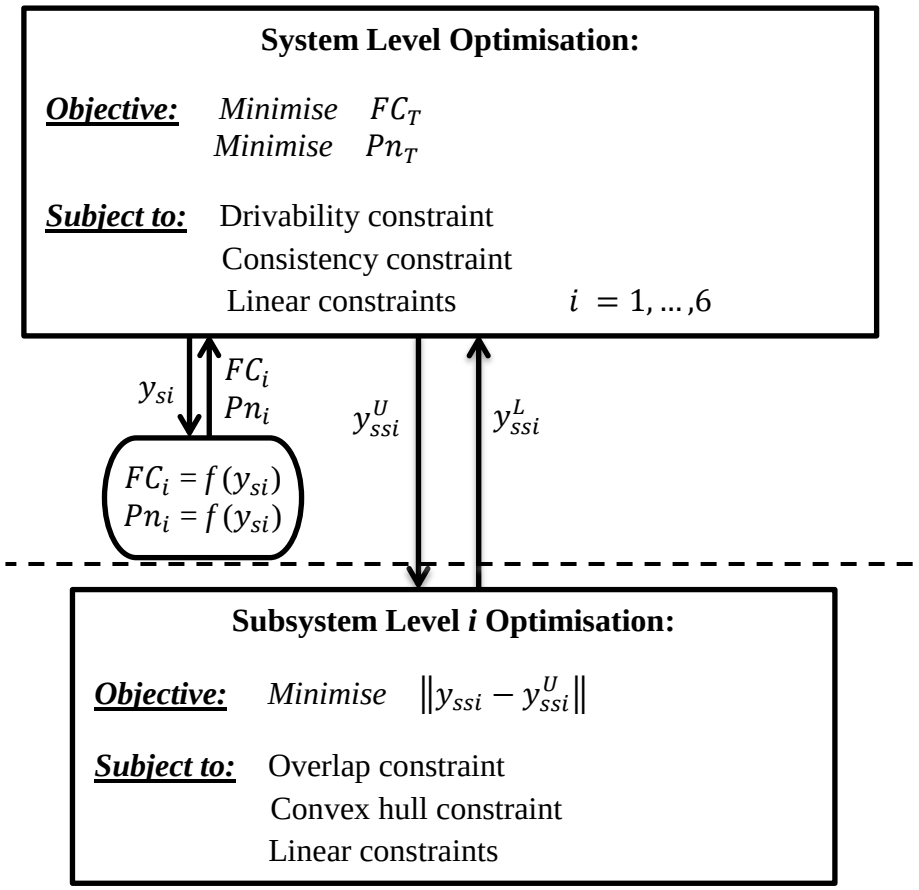


Figure 6.18: Decomposition and information flow of the CO process for GDI engine calibration optimisation problem

Equation 6.7 illustrates the CO formulation at the system level for the GDI engine calibration problem.

Objective:

$$\textit{Minimise} \quad f_0 = a \times \sum_{i=1}^{6} w_i \times FC\,(y_{si}) + b \times \sum_{i=1}^{6} w_i \times Pn\,(y_{si})$$

With respect to

$$(y_{si}, \hat{y}_{ssi}^L)$$

Subject to:

$$g_0 = Max\left(\left|y_{si} - y_{sj}\right|\right) - k \times \left|\mathrm{y}_{si}^{max} - \mathrm{y}_{si}^{min}\right| \leq 0$$

where i & $j = 1, \dots, 6$ $(i \neq j)$

$$h_0{}^c = \sum_{i=1}^{6} \left\|y_{si} - \hat{y}_{ssi}^L\right\| = 0$$

$$y_{si}^{min} \leq y_{si} \leq y_{si}^{max} \qquad i = 1, \dots, 6$$

Equation 6.7

f_0 is the system level objective function, w_i indicates the equivalent residency time for minimap *i* (given in Table 4.1), y_{si} is the vector of linking calibration parameters at the system level for discipline *i*, $\hat{y}_{ssi}^L$ is the copy of linking

calibration parameters transferred from subsystem level for discipline *i*, $FC\ (y_{si})$ and $Pn\ (y_{si})$ are the evaluated valued for fuel consumption and Pn by the analyser *i*, g_0 is the drivability constraint, ${h_0}^c$ defines the consistency constraint, and y_{si}^{min} and y_{si}^{max} are the lower and upper boundary limits for the linking variables.

In Equation 6.7, maximum allowable actuator change was set to half of design space (*k* = 0.5), and the '*a*' and '*b*' coefficients were set as: $a = 4$ and $b = 1$, similar to the MDF and 2-stage optimisation formulations (see Equation 6.6 and Equation 6.4).

Equation 6.8 illustrates the CO problem formulation at the subsystem level for each discipline *i*.

Objective:
Minimise $\|y_{ssi} - \hat{y}_{ssi}^U\|$

With respect to
y_{ssi}

Subject to:
$g_{ss1} = Overlap(y_{ssi}) \leq E_i$
$g_{ss2} = h(y_{ssi}) \leq 0$
$y_{ssi}^{min} \leq y_{ssi} \leq y_{ssi}^{max}$ Equation 6.8

In this equation, y_{ssi} is the vector of linking calibration parameters at the subsystem level for discipline *i*, $\hat{y}_{ssi}^U$ is the copy of linking calibration parameters transferred from the system level for discipline *i,* E_i is the maximum allowable valve overlap at discipline *i* (summarised in Table 6.1), and $h(y_{ssi})$ denotes the convex hull constraint at discipline *i*.

In this research, CO framework was also implemented for both calibration cases: calibration 'Case 1', over the single injection models (i.e. with 24 linking variables), and calibration 'Case 2', over the double injection models (i.e. with 32 linking variables).

6.3.2.2 CO Approach: Implementation

The CO framework for the GDI engine steady state calibration optimisation problem was implemented in MATLAB® programming environment. To implement the CO framework, firstly; all the required information was loaded into the program 'Main Body' (i.e. similar to the MDF implementation), including: linear constraints for the calibration parameters, boundary limits for the FC and Pn responses, speed / load operating conditions, residency weight for each minimap point over the NEDC drive cycle, and engine response models.

The XDSM diagram is used to explain how CO framework was programmed, as illustrated in Figure 6.19.

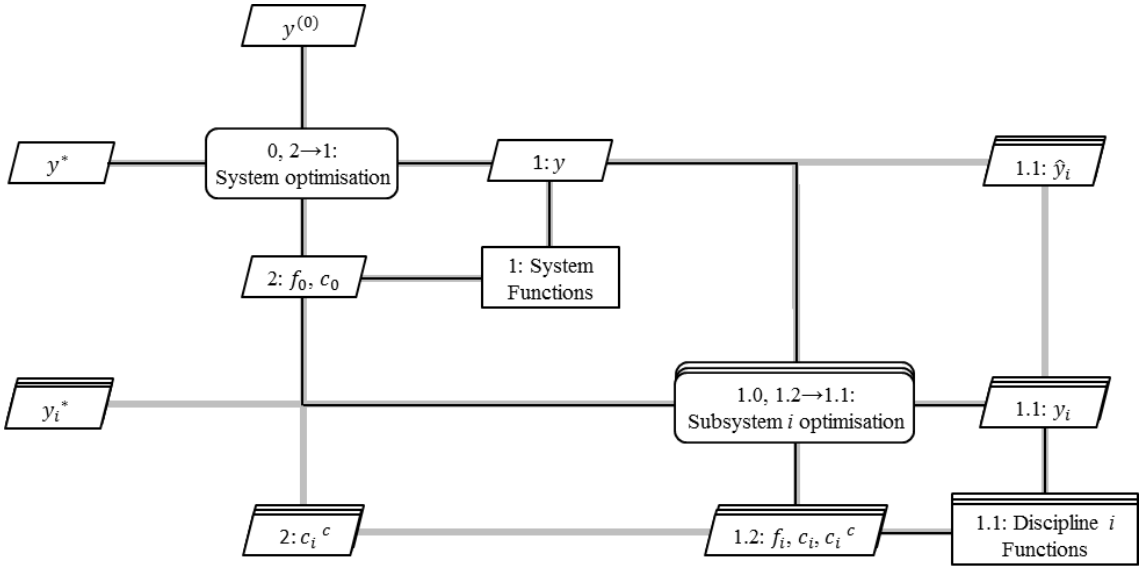


Figure 6.19: Illustration of CO strategy for solving the steady state engine calibration optimisation problems using the XDSM graph

For the system level optimisation, at **level 0**, the system optimiser starts the searching process from an initial search point $y^{(0)}$, which is transferred to the optimiser as an input. In CO implementation, the *fmincon* Matlab function [248] was used as the system level optimisation algorithm (i.e. similar to the MDF formulation). The optimisation algorithm transfers the system level optimum solution y into **level 1**, where the system level constraint c_0 (i.e. g_0) and objective function f_0 are evaluated for calibration setting y (see Equation 6.7).

Following, to define the consistency constraint, the system level calibration setting y is sent to **level 1.0**, to execute the subsystem level optimisation. At

level 1.0, the subsystem optimiser commences the search process from the cascaded calibration setting y. In the subsystem level optimisation, *fmincon* Matlab function [248] was used as the optimiser. The susceptibility of this gradient based search to be trapped in a local optimum is an advantage in the subsystem level optimisation, since it favours solutions close to the cascaded targets.

The subsystem level optimum solution y_i is then transferred to **level 1.1**, where the subsystem level objective function f_i (i.e. $\left\|y_{ssi} - \hat{y}_{ssi}^{U}\right\|$) and constraints c_i (i.e. g_{ss1} and g_{ss2}) are evaluated (see Equation 6.8). At **level 1.1**, the required information regarding the targets for each discipline *i* is available as an input $\hat{y}_i$. Next, the subsystem level solution is transferred to **level 1.2**, where the consistency constraint for each discipline *i* is defined ${c_i}^c$ (i.e. $\left\|y_{si} - \hat{y}_{ssi}^{L}\right\|$, shown in Equation 6.7). Subsequently, when the subsystem level optimisation process is terminated, the subsystem level optimum solution for each discipline ${y_i}^*$ is cascaded to the system level, **level 2**, to check whether the solution satisfies the system level convergence requirements. Thus, if the solution meets the convergence requirements, the optimisation algorithm terminates and the cascaded solutions from the subsystem level y^* is the final CO solution. Otherwise, the system optimiser uses the subsystem level solution as the initial search point for another iteration of the optimisation process.

The pseudocode for the developed CO framework, illustrated in Figure 6.19, is listed in Table 6.10.

Input: Initial linking variables ($y^{(0)}$)

Output: Optimal linking variables (y^*), objective function (f^*), constraint values (c^*)

0: Initiate the system level optimisation iteration

Repeat

1: Compute system level objective and constraint (f_0, c_0)

for each subsystem (discipline) *i* do

1.0: initiate subsystem optimisation

Repeat

1.1: Compute subsystem objective and constraint (f_i, c_i)

1.2: Compute new subsystem linking variables y_i and consistency constraint c_i^c

Until 1.2→1.1: Subsystem optimisation *i* has converged

end for

2: Compute new system design point

Until 2→1: System optimisation has converged

Table 6.10: CO Pseudo Code for Engine Applications

6.3.2.3 CO Approach: Results

I. Calibration Case 1

Given that *fmincon* optimisation algorithm was used at the system level, the CO framework was run with 5 random initial solutions (i.e. similar to MDF). The framework was converged to the optimum solution (i.e. best solution out of 5 runs) after 11 iterations in 3006 seconds. The CO optimisation solution for all the 24 variables at calibration 'Case 1' is summarised in Table 6.11. This table shows the corresponding FC (kg/hr) and Pn (pccm) for each engine speed / load operating condition, and the over cycle FC and Pn responses. The CO optimisation solutions are also illustrated in Figure 6.20 against the baseline setting (see Table 4.5) as a parallel axis plot at each minimap point, aiming to demonstrate the smoothness of the achieved optimisation solution compared to the benchmark setting (i.e. particularly for EVC and FRP calibration parameters).

Speed / Load (rpm – Nm)	IVO (ATDC)	EVC (ATDC)	FRP (MPa)	SOI (BTDC)	FC (kg/hr)	Pn (pccm)
700-28	31.7	15.1	12.3	294.5	1.376	4.53E+2
1500-41	31.7	11.5	13.1	296.1	3.205	6.18E+3
1250-125	1.7	36.5	13.5	308.3	4.679	4.79E+5
1500-105	1.7	36.5	11.0	322.5	5.082	1.24E+6
2000-81	4.6	36.5	14.0	300.8	5.859	7.45E+5
2000-199	1.7	36.5	11.0	302.7	10.741	2.81E+6
				Overall	2.734	2.03E+5

Table 6.11: CO calibration optimisation solution for the GDI engine calibration Case 1

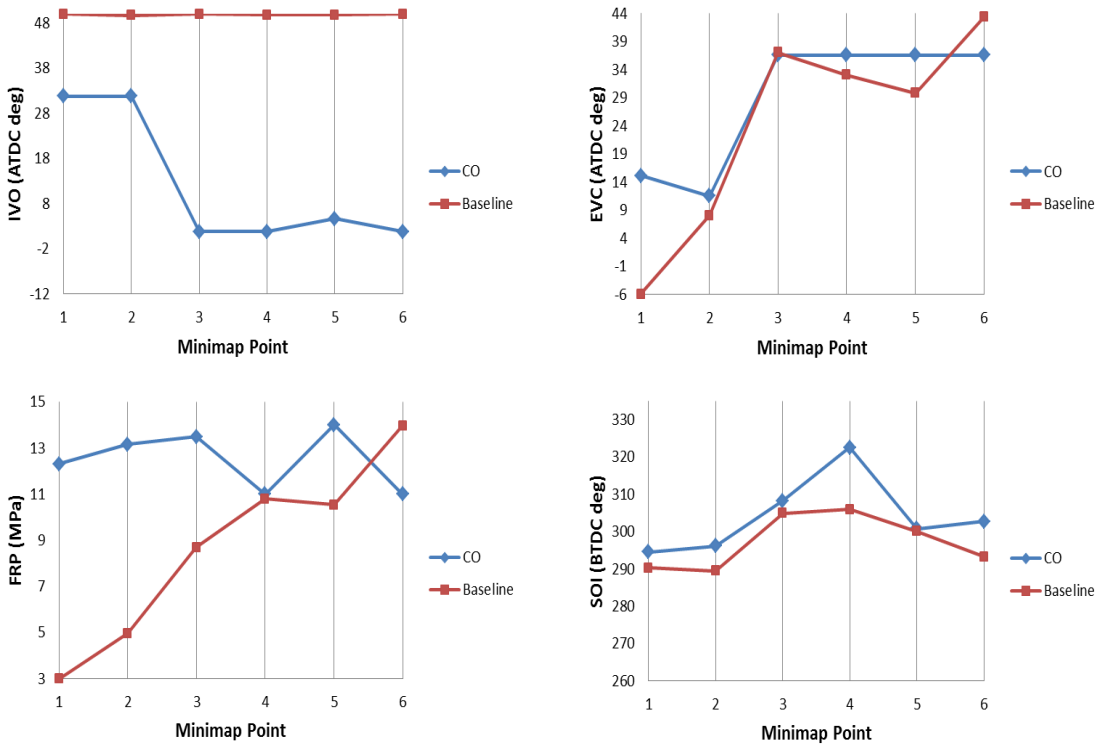


Figure 6.20: Actuator setting at each minimap for both baseline and CO approaches

Figure 6.21 and Figure 6.22 compare the CO optimisation solution versus the baseline calibration setting, in terms of percentage improvement in the overall fuel consumption and Pn. Accordingly, it can be seen that using CO as the optimisation framework enhanced the quality of calibration solution, by average improvement of 6.93% in FC and 62.3% in Pn.

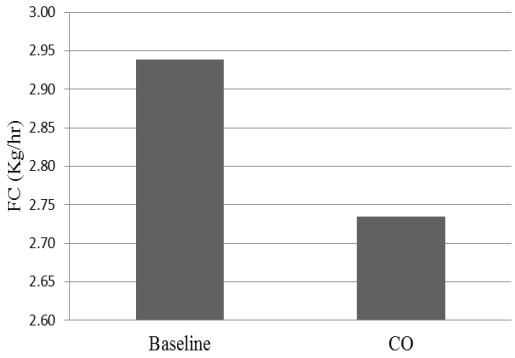


Figure 6.21: Comparing total FC over NEDC drive cycle between CO approach and baseline setting for calibration Case 1

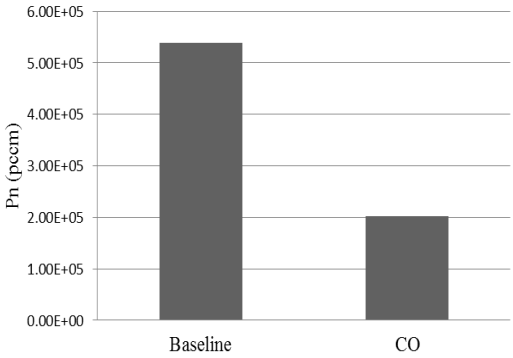


Figure 6.22: Comparing total Pn over NEDC drive cycle between CO approach and baseline setting calibration Case 1

II. Calibration Case 2

Similar to calibration Case 1, the CO framework was run with 5 random initial solutions. The *fmincon* optimisation algorithm converged into the optimum solution after 18 iterations in 18117 seconds. The CO optimisation solution for the GDI engine calibration problem Case 2 is illustrated in Table 6.12, including the optimum setting for all 32 calibration parameters, corresponding FC (kg/hr) and Pn (pccm) to each minimap point, and the total FC and Pn over the NEDC drive cycle. The CO optimisation solution is also illustrated against the benchmark setting in Figure 6.23, using a parallel axis plot. This figure shows that the CO framework could deliver a smooth engine map for all the calibration variables, through formulating the engine drivability requirements as a system constraint.

Minimap (rpm – Nm)	**IVO** (ATDC)	**EVC** (ATDC)	**FRP** (MPa)	**SOI1** (BTDC)	**Delay** (ms)	**Split** (%)	**FC** (kg/hr)	**Pn** (pccm)
700-28	41.5	18.1	12.0	293.7	NA	NA	1.373	2.30E+3
1500-41	32.7	11.5	13.1	295.6	NA	NA	3.205	6.03E+3
1250-125	19.2	36.5	14.0	312.8	4.9	0.32	4.701	8.34E+5
1500-105	49.2	36.4	13.2	285.0	2.7	0.48	5.065	1.14E+6
2000-81	46.5	34.2	12.0	313.1	4.0	0.38	5.922	1.98E+6
2000-199	19.2	36.5	13.2	336.3	2.4	0.37	10.719	2.17E+6
						Overall	2.735	2.56E+5

Table 6.12: CO calibration optimisation solution for the GDI engine calibration Case 2

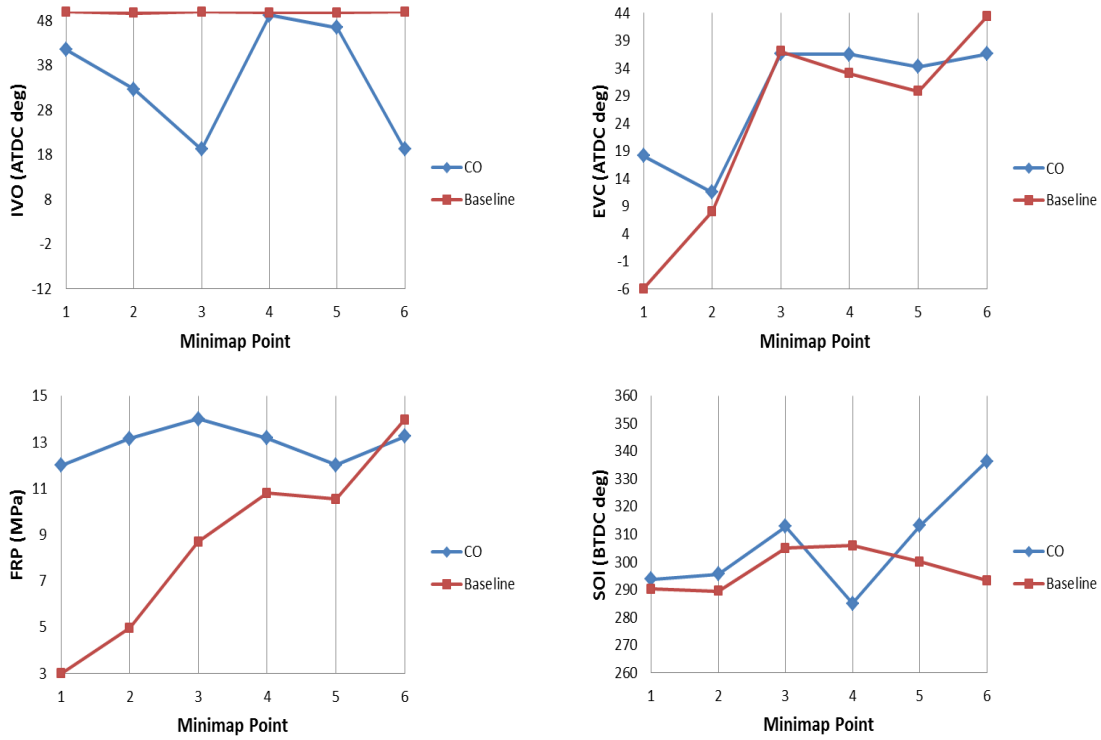


Figure 6.23: Actuator setting at each minimap for both baseline and CO approaches

Furthermore, the CO solution for calibration Case 2 was compared against the baseline calibration setting in terms of percentage improvement in the overall FC and Pn objectives, as shown in Figure 6.24 and Figure 6.25. These figures show that the CO approach outperforms the baseline calibration setting, improving the overall FC by 6.9% and overall Pn by 52.34%, respectively.

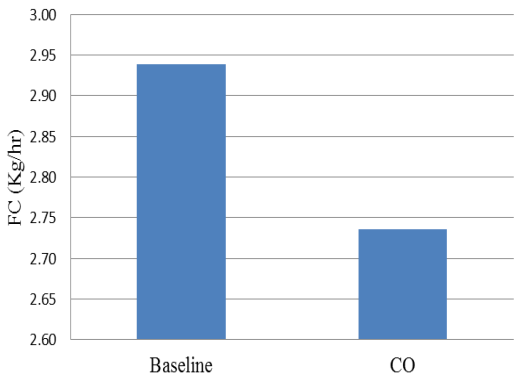


Figure 6.24: Comparing total FC over NEDC drive cycle between CO approach and baseline setting for calibration Case 2

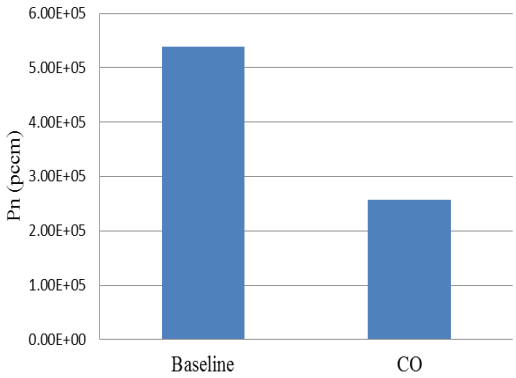


Figure 6.25: Comparing total Pn over NEDC drive cycle between CO approach and baseline setting for calibration Case 2

Moreover, comparing the CO optimisation solutions for calibration Cases 1 and 2, it was seen that the overall fuel consumption was approximately the same for both calibration cases, however; calibration Case 1 (i.e. over single injection models) produced less Pn over the NEDC drive cycle. This observation is consistent with the results attained from the other optimisation approaches (i.e. 2-stage and MDF).

6.3.3 Analytical Target Cascading Framework (ATC)

6.3.3.1 ATC Approach: Problem Analysis

The ATC optimisation architecture [42], [202], which is a multi-level hierarchal optimisation structure, was applied to formulate the GDI engine steady state calibration optimisation problem. In ATC optimisation framework, each optimisation level has its own design variables, objective function and design constraints (i.e. similar to the CO framework). The main difference between the ATC and CO approaches is the way system consistency is maintained. For the CO formulation, an equality constraint (i.e. consistency constraint) is responsible to maintain the whole system consistency, whereas the system consistency in ATC approach is ensured by minimising a penalty term ∅ at the system level. This penalty term is a function of the discrepancy between the cascaded target for the system level and the system level solution.

An ATC framework was associated with the engine calibration problem by treating each minimap point as a component of the *subsystem level optimisation* or discipline, and the engine performance over all minimaps (i.e. drive cycle) as the *system level optimisation* problem. Therefore, for the GDI engine case study, the calibration problem was decomposed into a bi-level ATC framework, consisting of a system level and a subsystem level with 6 disciplines. The nomenclature of the problem decomposition for each level of hierarchical optimisation process is detailed in Table 6.13.

Level	Decomposition of GDI engine calibration optimisation problem
System level *over a drive cycle*	*Objectives:* Minimise Overall fuel consumption (FC_T) Minimise Overall particulate number (Pn_T) Minimise Overall penalty function ($\emptyset_T$) *Constraints:* Drivability constraint Linear constraints *Design variables:* Calibration parameters over 6 minimap points
Interface	*Linking variables between system level and subsystem level i* Calibration parameters at subsystem level *i*
Subsystem level Discipline *i* *at each minimap point i*	*Objective:* Penalty function ($\emptyset_i$) *Constraints:* Overlap constraint Convex hull constraint Linear constraints *Design variables:* Calibration parameters at minimap *i*

Table 6.13: Decomposition of GDI engine calibration optimisation problem using ATC architecture

Figure 6.26 illustrates the organisation of the implemented ATC framework for the GDI engine calibration optimisation problem. Given that ATC is a target-based architecture, initially the optimisation targets for the linking variables y_s^T are sent to the system level optimisation. At the system level, the system optimisation algorithm optimises the system level objective function (i.e. consists of fuel consumption, Pn, and a penalty function) in subject to the system level constraint (i.e. drivability constraint). In the system level, the Pn and FC response values are evaluated by the response analysers. Afterwards, the system level optimisation output for each calibration parameter y_{ssi}^U is cascaded down to the corresponding discipline *i* at the subsystem level. At each discipline, the subsystem optimiser minimises the subsystem objective function, i.e. a penalty term which is a function of the discrepancy between the transferred calibration parameters from the system level y_{ssi}^U and the discipline calibration parameters y_{ssi}, in subject to the discipline constraints (i.e. overlap

and convex hull constraints). Subsequently, the subsystem level cascades the discipline solutions y_{ssi}^{L} back into the system level, as the new target for the system level optimisation. This iterative process continues until the set termination conditions at the system level are satisfied.

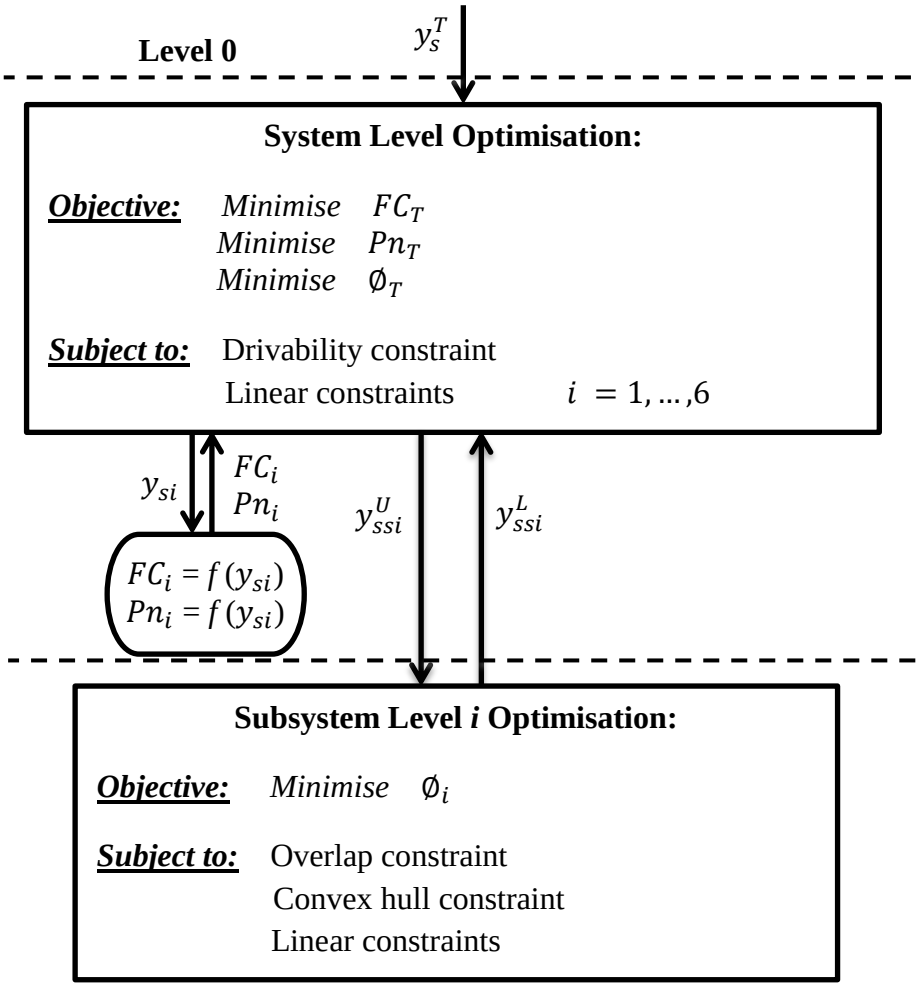


Figure 6.26: Decomposition and information flow of the ATC process for GDI engine calibration optimisation problem

In order to formulate the GDI engine problem using the ATC architecture, initially an auxiliary optimisation problem was solved prior to the ATC optimisation process, since no target value for the linking variables was available. The scope of this 'Level 0' optimisation was to define an initial target value for the linking variables at the system level of hierarchy. Therefore, a global optimisation algorithm (the standard Genetic Algorithm (GA) available in the Matlab Global Optimisation toolbox [249]) was employed for level 0 optimisation, to minimise the fuel consumption response, illustrated in Equation 6.9, over the range of calibration parameters without considering any constraint.

The weights w_i in Equation 6.9 are related to the equivalent residency time of the engine at each minimap in the NEDC emissions drive cycle (given in Table 4.1).

Objective:

Minimise $\sum_{i=1}^{6} w_i \times FC\ (y_i)$

With respect to

y_i

Subject to:

$y_i^{min} \leq y_i \leq y_i^{max} \qquad i\ = 1, \dots, 6$ Equation 6.9

The solution of level 0 optimisation (y_{si}^T) is then cascaded down to the system level as the initial target for the linking variables.

Equation 6.10 illustrates the ATC formulation at the system level for the GDI engine calibration problem. In this equation, the system level objective (R_S) was formulated as the weighted sum of FC and Pn responses, i.e. similar to the other MDO approaches formulated (see Equation 6.6 and Equation 6.7), plus a penalty function $\emptyset$ to minimise the inconsistency between levels of optimisation (i.e. system level and subsystem level). There are several types of penalty functions in literature [204]–[206], as discussed in Chapter 3. In this thesis, the augmented Lagrangian penalty function [206] was implemented, which enhances the ATC convergence property by updating the penalty weights until the desired consistency is achieved. In Equation 6.10, g_0 indicates the drivability constraint used to avoid the unacceptable calibration solutions with large actuator changes.

Objective:

Minimise $f_0 = R_s + \sum_{i=1}^{6} \emptyset_i (y_{si} - \hat{y}_{ssi}^L)$

$R_s = a\ \times \sum_{i=1}^{6} w_i \times FC\ (y_{si})\ +\ b\ \times \sum_{i=1}^{6} w_i \times Pn\ (y_{si})$

$\emptyset_i (y_{si} - \hat{y}_{ssi}^L) = v_i (y_{si} - \hat{y}_{ssi}^L)^T + \left\| W_i \text{ o } (y_{si} - \hat{y}_{ssi}^L) \right\|_2^2$

With respect to

$(y_{si}, \hat{y}_{ssi}^L)$

Subject to:

$g_0 = Max\left(\left|y_{si} - y_{sj}\right|\right) - k \times \left|y_{si}^{max} - y_{si}^{min}\right| \leq 0$

where $i\ \&\ j = 1, \dots, 6\ (i \neq j)$

$y_{si}^{min} \leq y_{si} \leq y_{si}^{max} \qquad i\ = 1, \dots, 6$ Equation 6.10

In Equation 6.10, y_{si} is the vector of linking calibration parameters at the system level for discipline *i*, $\hat{y}_{ssi}^{L}$ is the copy of linking calibration parameters transferred from subsystem level for discipline *i*, *v* is the vector of Lagrange multipliers, *W* is the vector of penalty weights, and the Hadamard symbol, shown by 'o', is used to denote term-by-term multiplication of vectors [250]. Besides, similar to other MDO formulations, '***a***' and '***b***' coefficients were set as: $a = 4$ and $b = 1$, and *k* was set as 0.5.

For the augmented Lagrangian penalty function [206], the penalty weights are updated at each optimisation iteration *q*, as illustrated in Equation 6.11.

$$W_i^{q+1} = \begin{cases} W_i^q \;\; if \;\; \left|(y_{si} - \hat{y}_{ssi}^{L})^q\right| \leq \gamma \left|(y_{si} - \hat{y}_{ssi}^{L})^{q-1}\right| \\ \beta W_i^q \;\; if \;\; \left|(y_{si} - \hat{y}_{ssi}^{L})^q\right| > \gamma \left|(y_{si} - \hat{y}_{ssi}^{L})^{q-1}\right| \end{cases} \qquad \text{Equation 6.11}$$

W_i^q indicates the penalty weight for linking variable *i* at iteration *q*, where β and γ are arbitrary coefficients to be defined, in a way that $\beta > 1$ and $0 < \gamma < 1$ [206]. In this thesis, the penalty function coefficients were set as: $\beta = 1.3$ and $\gamma = 0.1$, that are recommended in literature [251], [252], to converge to an optimum solution robustly.

Furthermore, the Lagrange multipliers v are updated iteratively as shown in Equation 6.12.

$$v_i^{q+1} = v_i^q + 2 \times W_i^q \; o \; W_i^q \; o \; (y_{si} - \hat{y}_{ssi}^{L})^q \qquad \text{Equation 6.12}$$

Equation 6.13 illustrates the ATC problem formulation at the subsystem level for each discipline *i*.

Objective:
Minimise $\quad \emptyset\left(y_{ssi} - \hat{y}_{ssi}^{U}\right)$

With respect to
y_{ssi}

Subject to:
$g_{ss1} = Overlap(y_{ssi}) \leq E_i$
$g_{ss2} = h(y_{ssi}) \leq 0$
$y_{ssi}^{min} \leq y_{ssi} \leq y_{ssi}^{max}$ $\qquad$ Equation 6.13

y_{ssi} is the vector of linking calibration parameters at the subsystem level for discipline *i*, $\hat{y}_{ssi}^{U}$ is the copy of linking calibration parameters cascaded down from the system level for discipline *i,* $\emptyset$ is the penalty function used to minimise the discrepancy between y_{ssi} and $\hat{y}_{ssi}^{U}$, E_i is the maximum allowable valve overlap at discipline *i* (summarised in Table 6.1), and $h(y_{ssi})$ denotes the convex hull constraint at discipline *i*. Noteworthy, there is no direct coupling between disciplines in the subsystem levels as they do not share any of the variables.

The ATC framework was implemented for both calibration cases: calibration 'Case 1', over the single injection models (i.e. with 24 linking variables), and calibration 'Case 2', over the double injection models (i.e. with 32 linking variables).

6.3.3.2 ATC Approach: Implementation

The ATC framework for the GDI engine case study was implemented in MATLAB® programming environment. In this implementation, the required information about the case study was loaded into the program 'Main Body' prior to the optimisation process (i.e. similar to CO and MDF implementation), containing: linear constraints for the calibration parameters, boundary limits for the FC and Pn responses, speed / load operating conditions, residency weight for each minimap point over the NEDC drive cycle, and engine response models.

The XDSM diagram is used to explain the way ATC framework was implemented for the GDI engine case study, as illustrated in Figure 6.27.

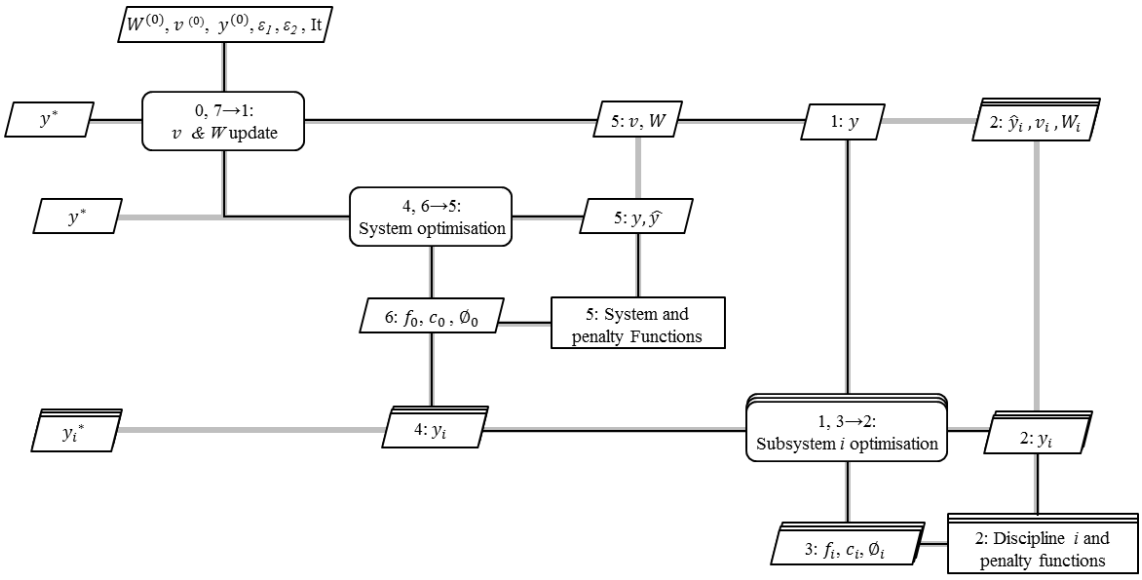


Figure 6.27: Illustration of ATC strategy for solving the steady state engine calibration optimisation problems using the XDSM graph

In this implementation, first of all, a number of variables were defined for the ATC framework, which are shown as inputs at **level 0** (i.e. $W^{(0)}$, $v^{(0)}$, $y^{(0)}$, ε_1, ε_2, It). $W^{(0)}$ defines the initial vector for the penalty weights, $v^{(0)}$ is the initial vector of Lagrange multipliers, $y^{(0)}$ is the initial target for the linking variables, ε_1 and ε_2 are the termination conditions which are introduced to terminate the iterative coordination algorithm between system level and subsystems [252] (given in Equation 6.14), and *It* defines the maximum number of optimisation iterations as a termination condition for the ATC framework.

$$\|d^k - d^{k-1}\|_\infty < \varepsilon_1$$
$$\|d^k\|_\infty < \varepsilon_2 \qquad \text{Equation 6.14}$$

d^k is a vector which denotes the collection of discrepancies between the system level target and subsystem level solution $(y_{si} - y_{ssi}^L)$ for all the linking variables at iteration *k*.

For the GDI engine case study, $W^{(0)}$ was set as a vector of ones, $v^{(0)}$ was set as a vector of zeros, It was set as 100, ε_1 and ε_2 were set as $\varepsilon_1 = \varepsilon_2 = 0.01$, and $y^{(0)}$ was defined by solving the auxiliary optimisation problem given in Equation 6.9 prior to the ATC process. To solve the auxiliary optimisation problem, the Matlab GA algorithm [249] was run for 200 iterations while the

'Population Size' was 200, 'Crossover Rate' was 0.8, 'Mutation Rate' was 0.05, and 'Elite Size' was 5.

The optimisation process commences at **level 0**, by sending the target calibration solution y into the subsystem level optimisation, i.e. **level 1**. In this level, the subsystem analyser starts the optimisation process from the cascaded initial search point y, and sends the subsystem level solution y_i into **level 2** to evaluate the subsystem level objective f_i, constraint c_i (i.e. g_{ss1} and g_{ss2}), and penalty function $\emptyset_i$ (see Equation 6.13). The required information at each discipline *i* to compute the objective and penalty functions (i.e. the optimisation targets ($\hat{y}_i$), penalty weights (W_i), and Lagrange multipliers (v_i)) are transferred to **level 2** as inputs. Following, at **level 3**, the optimum linking variables for each discipline y_i are updated. Then, when the subsystem level optimisation process is terminated, the optimum solution y_i^* is transferred to the system level optimisation, i.e. **level 4**, where the system optimiser begins the optimisation process using the subsystem level optimum solution as the initial search point y_i. The system optimiser sends the system level solution y and a copy of the cascaded subsystem level solutions $\hat{y}$ into **level 5**, to evaluate the system level objective function f_0, constraints c_0 (i.e. g_0), and penalty function $\emptyset_0$ (see Equation 6.10). After that, at **level 6** the system level optimum solution y is updated. Subsequently, when the system level optimisation process is terminated, the optimum solution y^* is cascaded to **level 7**, to check whether the optimum solution fulfils the ATC framework convergence criteria ($\varepsilon_1,\ \varepsilon_2,\ It$). Thus, if the solution meets the convergence requirements, the optimisation process terminates and the cascaded solution from the system level y^* is the final ATC solution. Otherwise, the vectors of penalty weights (*W*) and Lagrange multipliers (*v*) are updated and the system level optimum solution y^* is sent to **level 1** as the new initial target for the next iteration of the optimisation process.

In ATC implementation, the *fmincon* Matlab function [248] was employed to solve the optimisation problems at both system and subsystem levels, since the susceptibility of this gradient based search to be trapped in a local optimum close to the cascaded targets is an advantage for both levels of optimisation.

The pseudocode for the developed ATC framework, illustrated in Figure 6.27, is listed in Table 6.14.

Input: Initial linking variables ($y^{(0)}$), ATC algorithm variables ($W^{(0)}$, $v^{(0)}$, ε_1, ε_2, It)
Output: Optimal variables (x^*), objective function (f^*), constraint values (c^*)
0: Initiate ATC iteration
Repeat
for each subsystem (discipline) *i* do
1: initiate subsystem optimisation
Repeat
2: Compute subsystem objective, constraint and penalty function (f_i, c_i, $\emptyset_i$)
3: Update subsystem linking variables
Until 3→2: Subsystem optimisation *i* has converged
end for
4: initiate system optimiser
Repeat
5: Compute system level objective, constraint, and penalty function (f_0, c_0, $\emptyset$)
6: Update system linking variable copies
Until 6→5: System optimisation has converged
7: Update penalty weights (v, W)
Until 7→1: System convergence criteria are fulfilled (It, ε_1, ε_2)

Table 6.14: ATC Pseudo Code for Engine Applications

6.3.3.3 ATC Approach: Results

I. Calibration Case 1

The ATC framework was run with 5 random initial solutions (i.e. similar to MDF and CO frameworks), since the *fmincon* optimisation algorithm was used as the system level optimiser. The ATC framework converged into a single solution after 8 iterations in 224 seconds, while the termination constraints ε_1 and ε_2 (Equation 6.14) was set as $\varepsilon_1 = \varepsilon_2 = 0.01$. The optimum calibration solution for the GDI engine calibration Case 1 is illustrated in Table 6.15. This table summarises the optimum setting for each of the 24 calibration parameters, using the ATC optimisation framework, and the corresponding calibration objectives, i.e. FC (kg/hr) and Pn (pccm), for each engine minimap point. Furthermore, the total FC and Pn responses over the reference drive cycle (i.e. NEDC drive cycle) are also shown in Table 6.15.

Speed / Load (rpm – Nm)	**IVO** (ATDC)	**EVC** (ATDC)	**FRP** (MPa)	**SOI** (BTDC)	**FC** (kg/hr)	**Pn** (pccm)
700-28	30.8	22.5	11.2	332.2	1.358	9.85E+3
1500-41	29.8	24.4	11.1	311.6	3.192	6.85E+3
1250-125	2.0	31.8	11.6	307.3	4.721	6.49E+5
1500-105	3.2	34.3	10.1	302.6	5.114	1.12E+6
2000-81	0.2	14.2	8.4	298.0	5.900	1.75E+6
2000-199	0.3	39.4	11.1	300.8	10.730	3.07E+6
				Overall	2.729	2.34E+5

Table 6.15: ATC calibration optimisation solution for the GDI engine calibration Case 1

Figure 6.28 illustrates both ATC solution and the baseline setting for the calibration variables at each minimap. It can be seen that the ATC framework delivered a smooth engine map for all the calibration variables by limiting the maximum allowable change for each variable over the transfers between the minimaps (k = 0.5).

Figure 6.29 and Figure 6.30 show the performance of the ATC optimisation framework versus the baseline calibration setting, in terms of percentage improvement for both the overall FC and Pn objectives. Compared to the baseline setting benchmark, the derived calibration setting using the ATC

optimisation approach delivers a drive cycle average enhancement in FC of 7.11 %, and a reduction in Pn of 56.46 %.

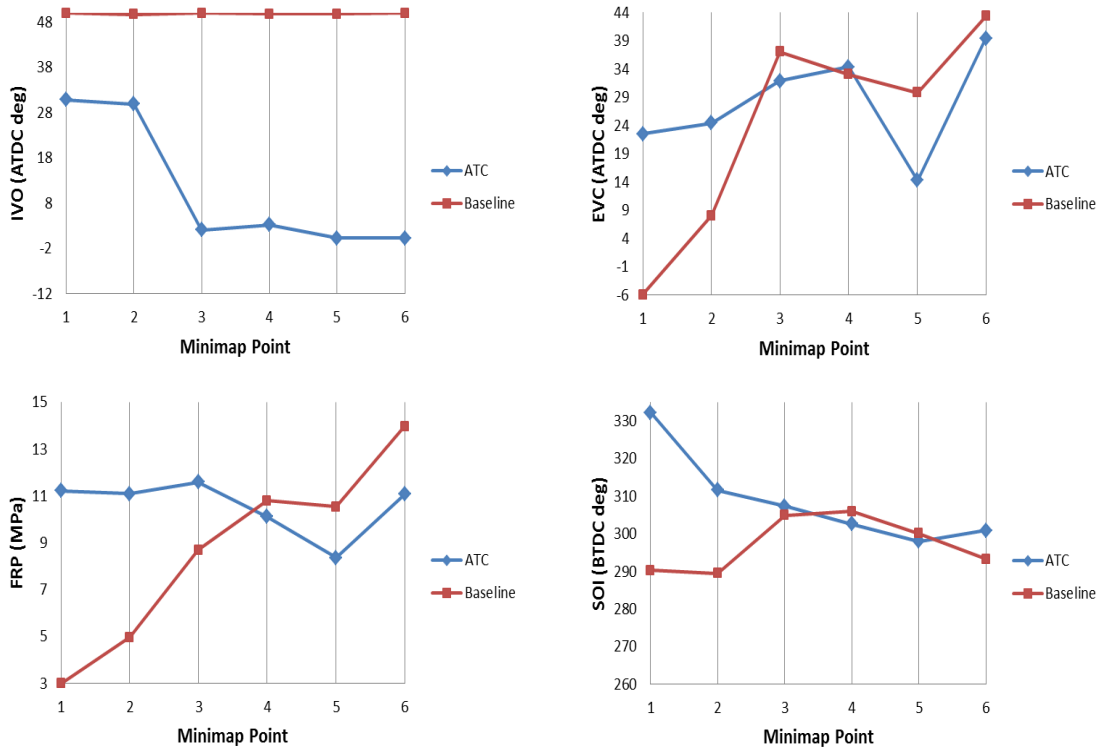


Figure 6.28: Actuator setting at each minimap for both baseline and ATC approaches

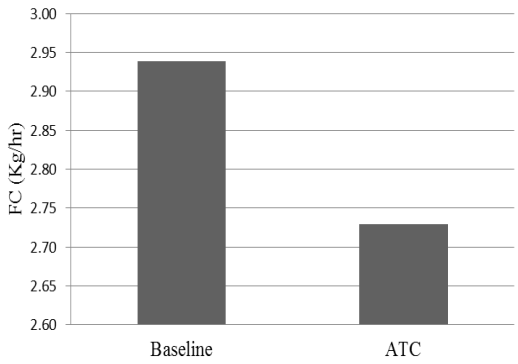


Figure 6.29: Comparing total FC over NEDC drive cycle between ATC approach and baseline setting for calibration Case 1

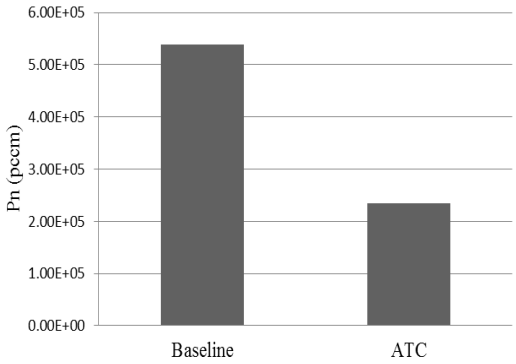


Figure 6.30: Comparing total Pn over NEDC drive cycle between ATC approach and baseline setting for calibration Case 1

II. Calibration Case 2

Similar to calibration Case 1, the ATC framework was run with 5 random initial solutions. The ATC framework converged into the optimum solution after 7 iterations in 1969 seconds, while termination constraints was set as $\varepsilon_1 = \varepsilon_2 =$

0.01. The ATC solution for calibration Case 2 is illustrated in Table 6.16. The corresponding FC (kg/hr) and Pn (pccm) responses to each minimap point, and the total FC and Pn responses over the NEDC drive cycle are also summarised in this table.

Minimap (rpm – Nm)	**IVO** (ATDC)	**EVC** (ATDC)	**FRP** (MPa)	**SOI1** (BTDC)	**Delay** (ms)	**Split** (%)	**FC** (kg/hr)	**Pn** (pccm)
700-28	49.2	23.5	11.1	295.8	NA	NA	1.334	9.66E+3
1500-41	49.2	25.0	11.0	324.5	NA	NA	3.133	1.26E+4
1250-125	22.7	42.4	13.1	320.3	4.3	0.51	4.688	7.61E+5
1500-105	22.7	37.3	13.4	342.7	7.4	0.55	5.074	1.02E+6
2000-81	46.8	43.3	13.4	305.4	4.2	0.61	5.852	1.87E+6
2000-199	22.7	43.3	13.2	331.3	3.4	0.47	10.691	2.19E+6
						Overall	2.692	2.45E+5

Table 6.16: ATC calibration optimisation solution for the GDI engine calibration Case 2

Figure 6.31 illustrates the calibration parameters (i.e. IVO, EVC, FRP and SOI) for both ATC solution and the baseline setting at each minimap point using a parallel axis plot. It can be seen that the ATC framework delivered a smooth engine map for all the calibration variables by limiting the maximum allowable change for each variable, i.e. Equation 6.10.

Furthermore, the performance of ATC optimisation framework for calibration Case 2 is shown in Figure 6.32 and Figure 6.33, in terms of percentage improvement for both the overall FC and Pn responses. The improvements over the drive cycle show that the ATC approach outperforms the baseline calibration setting for both overall FC and Pn responses, which were improved by 8.37 % and 54.52 %, respectively.

Comparing the overall FC and Pn responses achieved from calibration Case 1 (i.e. single injection strategy) and calibration Case 2 (i.e. double injection strategy) shows that using double injection strategy for the GDI engine reduced the overall fuel consumption, while increased the overall Pn marginally. Therefore, double injection strategy does not result in Pn reduction for the GDI engine within the tested operating conditions, which is also consistent with the optimisation results delivered by the other optimisation strategies applied in this research (i.e. 2-stage, MDF and CO). However, considering that ATC

optimisation solution for calibration Case 2 shows improvements in overall fuel consumption compared to calibration Case 1 (i.e. 1.26%), it can be argued that by applying an efficient optimisation strategy, double injection strategy can reduce the overall fuel consumption for the GDI engine case study.

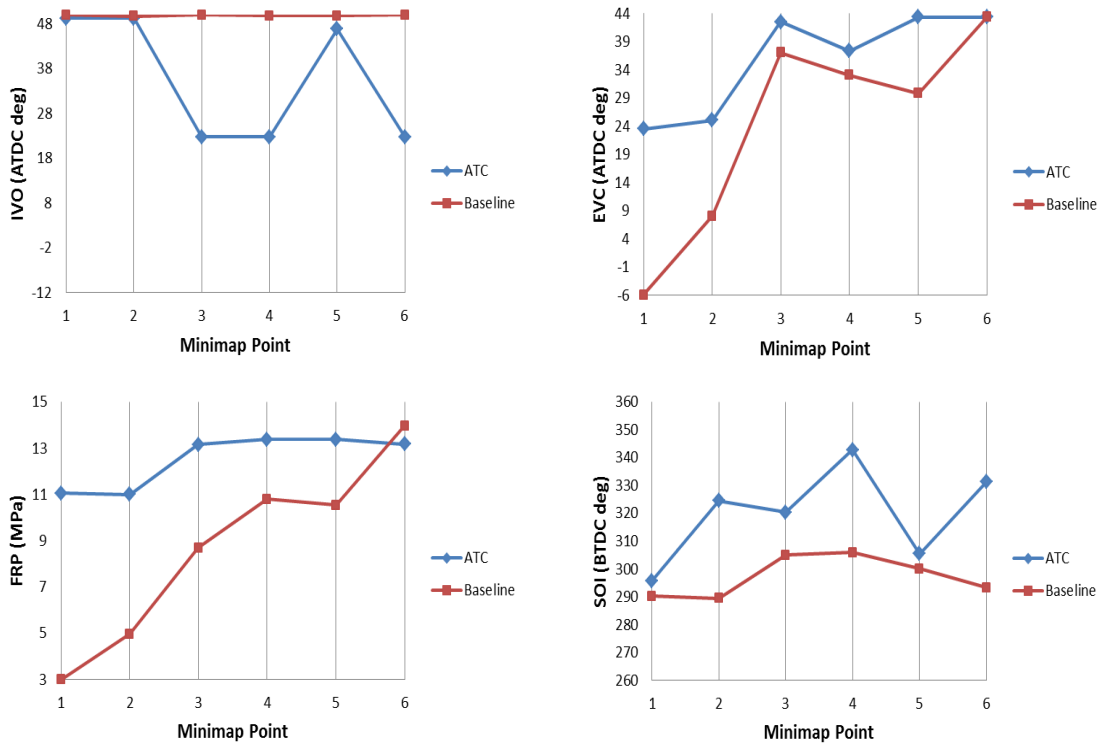


Figure 6.31: Actuator setting at each minimap for both baseline and ATC approaches

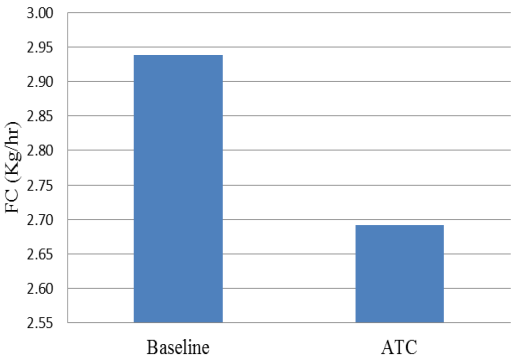


Figure 6.32: Comparing total FC over NEDC drive cycle between ATC approach and baseline setting for calibration Case 2

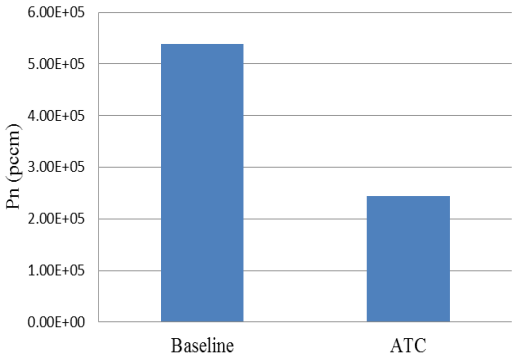


Figure 6.33: Comparing total Pn over NEDC drive cycle between ATC approach and baseline setting for calibration Case 2

6.3.4 Discussion of MDO Results for the GDI Engine Case Study

The performance of the optimisation frameworks applied for the GDI engine case study are summarised in Table 6.17 and Table 6.18 for the implemented calibration strategies. In this table, the advantages of each optimisation framework compared to the baseline solution are expressed in terms of percentage improvement in overall fuel consumption and Pn, and also the computational timing. It should be noted that the whole optimisation process for the 2-stages approach was executed within 2 days.

	FC (kg/hr)	FC (%)	Pn (pccm)	Pn (%)	Time (sec)
Baseline	2.940	---	5.38E+5	---	---
2-Stage	2.737	*6.82*	1.36E+5	*74.79*	---
MDF	2.734	*6.95*	2.06E+5	*61.72*	179
CO	2.734	*6.93*	2.03E+5	*62.30*	3006
ATC	2.729	*7.11*	2.34E+5	*54.46*	224

Table 6.17: Comparison between the calibration solutions delivered by different optimisation frameworks for the GDI engine calibration Case 1

	FC (kg/hr)	FC (%)	Pn (pccm)	Pn (%)	Time (sec)
Baseline	2.940	---	5.38E+5	---	---
2-Stage	2.730	*7.13*	1.97E+5	*63.35*	---
MDF	2.740	*6.87*	2.34E+5	*56.41*	216
CO	2.740	*6.90*	2.56E+5	*52.34*	18117
ATC	2.690	*8.37*	2.45E+5	*54.52*	1969

Table 6.18: Comparison between the calibration solutions delivered by different optimisation frameworks for the GDI engine calibration Case 2

Table 6.17 and Table 6.18 show that the ATC optimisation strategy outperformed the other optimisation approaches in terms of delivering improvements in FC, while it also delivered a reasonable enhancement in particulates. It should be underlined that superiority of the ATC framework to the other optimisation approaches was more significant for calibration ‘Case 2’, which was a more complex optimisation problem (i.e. with more optimisation variables). The MDF and CO optimisation approaches could also reduce the overall fuel consumption and particulates, however; the MDF approach converged to an optimum solution faster than CO.

Moreover, after comparing the overall FC and Pn responses achieved from calibration Cases 1 and 2, it was concluded that for the GDI engine case study at the tested minimap points double injection strategy was not beneficial to

reduce the overall particulates. However, it was observed that double injection strategy would result in a notable FC reduction over the drive cycle through application of an efficient calibration optimisation process (i.e. ATC).

Overall, the main limitation of the GDI engine case study was that the dimensionality of the calibration problem was small compared to a typical base calibration, which is normally executed for a full engine speed / load space with 10 to 30 minimap points and more complex constraints for the calibration. So, the developed MDO frameworks were also used to optimise the steady state calibration problem for an alternative case study based on a diesel engine. In the next section, the problem formulation and optimisation results for the diesel engine case study are presented.

6.4 Further Validation of Multidisciplinary Design Optimisation Frameworks for the Steady State Engine Calibration Optimisation Problems

As discussed in Chapter 4, the diesel engine calibration problem was to minimise the fuel consumption over the NEDC drive cycle subject to the emission constraints on the overall NOx and Pm responses (see Table 4.10). Moreover, each engine speed / load operating point was subjected to a constraint on combustion noise, which was set to control the maximum noise at each minimap point (see Table 4.11).

In order to demonstrate the complexity of diesel engine calibration optimisation problem, a feasibility measure (P_f) was used to measure the probability that a randomly generated solution (x), within the boundary limits for the calibration parameters, is within the feasible region of the diesel engine (F).

$$P_f = p\{x \in F : x^l \leq x \leq x^u\} \qquad \text{Equation 6.15}$$

Therefore, if N_f is the number of feasible solutions out of N randomly and uniformly generated solutions within the domain of the problem, the feasibility P_f can be estimated as the ratio of the feasible solutions N_f to the total number of generated solutions N:

$$P_f \approx \frac{N_f}{N} \qquad \text{Equation 6.16}$$

To study the complexity of the optimisation problem for the diesel engine case study, 10^5 random solutions were generated within the calibration domain using an OLH design [253]. The reason for using an OLH design, which is a space-filling DoE strategy, was to effectively explore design space over all parameters domain [65]. The next step was to apply all the calibration constraints, shown in Table 4.10 and Table 4.11, to the random solutions in order to define how many of them can meet all the constraints, as given in Equation 6.17.

$$\sum_i^{11} w_i \times Pm_i \leq 3$$
$$\sum_i^{11} w_i \times NOx_i \leq 6.9$$
$$Noise_i \leq E_i \qquad i = 1, \dots, 11$$

Equation 6.17

Overall, it was observed that only 16 solutions out of the 10^5 random solutions could meet all the optimisation constraints. The probability of finding a feasible solution was then calculated using Equation 6.16, to be 0.00016. It means that the probability of a randomly generated solution for the diesel engine calibration optimisation problem to be feasible is **0.016 %**, which clearly indicates the necessity of employing an efficient optimisation framework for this optimisation problem to deliver an optimum solution. Table 6.19 shows the probability (%) that a randomly solution meet each of the constraints given in Equation 6.17. It can be concluded from this table that NOx constraint is the hardest to satisfy.

Constraint	Initial Pop	Feasible	Probability (%)
Pm	100,000	93679	93.679
NOx	100,000	643	0.643
Noise	100,000	894	0.894
Pm-NOx-Noise	100,000	16	**0.016**

Table 6.19: Diesel engine constraint analysis

6.4.1 Problem Analysis

6.4.1.1 Multidisciplinary Feasible (MDF) Approach

The diesel engine calibration problem was associated with the MDF framework by formulating the engine performance over all 11 minimap points as the *system level optimisation*, while treating each minimap point as a *subsystem*

analyser. Accordingly, the nomenclature of the MDF architecture for the diesel engine calibration problem can be detailed as illustrated in Table 6.20.

Level	Decomposition of diesel engine calibration problem
System level Optimisation *over a drive cycle*	*Objectives:* Minimise Overall fuel consumption *Constraints:* Overall Pm Overall NOx Noise constraint at each minimap point Convex hull constraint at each minimap point Linear constraints *Design variables:* Calibration parameters over 11 minimap points

Table 6.20: Diesel engine calibration optimisation problem using MDF

The MDF formulation for the diesel engine calibration problem is illustrated in Equation 6.18.

Objective:

Minimise $f_0 = \sum_{i=1}^{11} w_i \times FC\ (y_{si})$

With respect to

y_{si}

Subject to:

$$g_{01} = \sum_{i}^{11} w_i \times Pm\ (y_{si}) \leq 3$$
$$g_{02} = \sum_{i}^{11} w_i \times NOx\ (y_{si}) \leq 6.9$$
$$g1_i = Noise(y_{si}) \leq E_i \qquad i = 1, \dots, 11$$
$$g2_i = h(y_{si}) \leq 0 \qquad i = 1, \dots, 11$$
$$y_{si}^{min} \leq y_{si} \leq y_{si}^{max} \qquad i = 1, \dots, 11$$

Equation 6.18

f_0 is the system level objective function, w_i indicates the equivalent residency time for minimap *i* (given in Table 4.6), y_{si} denotes the vector of linking calibration parameters at minimap *i* (given in Table 4.9), $FC\ (y_{si})$, $Pm\ (y_{si})$, $NOx\ (y_{si})$, and $Noise\ (y_{si})$ are the predicted engine responses at subsystem analyser *i*, g_{01} is the overall Pm constraint, g_{02} is the overall NOx constraint, $g1_i$ defines the noise constraint at minimap *i*, $g2_i$ indicates the convex hull constraint at minimap *i*, and y_{si}^{min} and y_{si}^{max} are the lower and upper boundary limits for the linking variables. In this equation, E_i defines the upper limit for Noise at minimap point *i*, summarised in Table 4.11. Also, $h(y_{si}) \leq 0$, ensures

that the calibration solution is within the convex hull envelope. The convex hull constraint was introduced as an additional design space constraint for this case study, i.e. using the Matlab convex hull function, to avoid extrapolation in parts of design space unexplored through testing.

6.4.1.2 Collaborative Optimisation (CO) Approach

For the diesel engine case study, the calibration problem was decomposed into a CO framework, consisting of a system level and a subsystem level with 11 disciplines. Accordingly, the nomenclature of the diesel engine problem decomposition is detailed in Table 6.21.

Level	Decomposition of diesel engine calibration problem
System level *over a drive cycle*	*Objectives:* Minimise Overall fuel consumption *Constraints:* Overall Pm Overall NOx Consistency constraint Linear constraints *Design variables:* Calibration parameters over 11 minimap points
Interface	*Linking variables between system level and subsystem i*: Calibration parameters at subsystem level *i*
Subsystem level Discipline *i* *at each minimap point i*	*Objective:* Minimise the discrepancy to the cascaded target *Constraints:* Noise constraint Convex hull constraint Linear constraints *Design variables:* Calibration parameters at minimap *i*

Table 6.21: Decomposition of diesel engine calibration optimisation problem using CO architecture

Equation 6.19 illustrates the CO formulation at the system level for the diesel engine calibration problem.

Objective:

Minimise $f_0 = \sum_{i=1}^{11} w_i \times FC\ (y_{si})$

With respect to

$(y_{si}, \hat{y}_{ssi}^{L})$

Subject to:

$g_{01} = \sum_{i}^{11} w_i \times Pm\ (y_{si}) \leq 3$

$g_{02} = \sum_{i}^{11} w_i \times NOx\ (y_{si}) \leq 6.9$

$h_0{}^{c} = \sum_{i=1}^{11} \left\| y_{si} - \hat{y}_{ssi}^{L} \right\| = 0$

$y_{si}^{min} \leq y_{si} \leq y_{si}^{max} \qquad i = 1, \ldots, 11$

Equation 6.19

y_{si} is the vector of linking calibration parameters at the system level for discipline *i*, $\hat{y}_{ssi}^{L}$ is the copy of linking calibration parameters transferred from subsystem level for discipline *i*, and $h_0{}^{c}$ defines the consistency constraint.

Then, at the subsystem level, the diesel engine optimisation problem was formulated as:

Objective:

Minimise $\left\| y_{ssi} - \hat{y}_{ssi}^{U} \right\|$

With respect to

y_{ssi}

Subject to:

$g_{ss1} = Noise(y_{ssi}) \leq E_i$

$g_{ss2} = h(y_{ssi}) \leq 0$

$y_{ssi}^{min} \leq y_{ssi} \leq y_{ssi}^{max}$

Equation 6.20

In this equation, y_{ssi} is the vector of linking calibration parameters at the subsystem level for discipline *i*, and $\hat{y}_{ssi}^{U}$ is the copy of linking calibration parameters transferred from the system level for discipline *i.*

Overall, the diesel engine calibration optimisation problem has 58 linking variables between levels of optimisation, but there is no direct coupling between subsystems as they do not share any of the variables.

6.4.1.3 Analytical Target Cascading (ATC) Approach

The diesel engine calibration problem was structured as a bi-level ATC framework, consisting of a system level and a subsystem level with 11 disciplines. The nomenclature of the problem decomposition for each level of hierarchical optimisation process is detailed in Table 6.22.

Level	Decomposition of diesel engine calibration problem
System level *over a drive cycle*	*Objectives:* Minimise Overall fuel consumption Minimise Overall penalty function *Constraints:* Overall Pm Overall NOx Linear constraints *Design variables:* Calibration parameters over 11 minimap points
Interface	*Linking variables between system level and subsystem i*: Calibration parameters at subsystem level *i*
Subsystem level Discipline *i* *at each minimap point i*	*Objective:* Penalty function *Constraints:* Noise constraint Convex hull constraint Linear constraints *Design variables:* Calibration parameters at minimap *i*

Table 6.22: Decomposition of diesel engine calibration optimisation problem using ATC architecture

Similar to the GDI engine case study, an auxiliary target setting optimisation problem was solved for the diesel engine case study prior to the ATC development. The standard Genetic Algorithm (GA), available in the Matlab Global Optimisation toolbox [249], was employed for this optimisation problem (**level 0**), formulated as unconstrained minimisation of the global fuel consumption response over the range of calibration variables, illustrated in Equation 6.21.

Objective:

Minimise $\sum_{i=1}^{11} FC(y_i)$

With respect to

y_i

Subject to:

$y_i^{min} \leq y_i \leq y_i^{max} \qquad i = 1, \ldots, 11$ Equation 6.21

The solution of level 0 optimisation was then cascaded down to the 'system' level as the initial target for the linking variables. The system level optimisation problem for the diesel engine using the augmented Lagrangian penalty function (∅) [206] was formulated in Equation 6.22.

Objective:

Minimise $\sum_{i=1}^{11} w_i \times FC\ (y_{si})\ + \sum_{i=1}^{11} \emptyset(y_{si} - \hat{y}_{ssi}^{L})$

With respect to

$(y_{si}, \hat{y}_{ssi}^{L})$

Subject to:

$g_{01} = \sum_{i=1}^{11} w_i \times Pm\ (y_{si}) \leq 3$

$g_{02} = \sum_{i=1}^{11} w_i \times NOx\ (y_{si}) \leq 6.9$

$y_{si}^{min} \leq y_{si} \leq y_{si}^{max} \qquad i = 1, \ldots, 11$ Equation 6.22

For the diesel engine case study, the penalty function coefficients, explained in Equation 6.11, were set as: $\beta = 1.3$ and $\gamma = 0.1$, similar to the GDI engine case study.

Equation 6.23 illustrates the ATC problem formulation at the subsystem level for each discipline *i*.

Objective:

Minimise $\emptyset\left(y_{ssi} - \hat{y}_{ssi}^{U}\right)$

With respect to

y_{ssi}

Subject to:

$g_{ss1} = Noise(y_{ssi}) \leq E_i$

$g_{ss2} = h(y_{ssi}) \leq 0$

$y_{ssi}^{min} \leq y_{ssi} \leq y_{ssi}^{max}$ Equation 6.23

6.4.2 Implementation and Results

For the diesel engine case study all the required engine response models were provided by the sponsoring company in CAMEO software environment. The first task was to export the engine response models into the developed MDO frameworks. CAMEO software is compatible with the Matlab software, so the engine models were exported into the Matlab workspace and loaded into the MDO frameworks using the 'data structure' format. Then, the developed MDO frameworks for the GDI engine were extended to accommodate the diesel engine objectives and constraints over the 11 minimap points. Similar to the GDI engine case study, the *fmincon* optimisation algorithm was used to solve the developed MDO frameworks (i.e. at both system and subsystem levels). Given that the performance of the gradient based algorithms can be affected by the location of algorithm initial search point, the MDO frameworks were run with 5 random initial solutions and subsequently the best solution was considered as the optimum solution.

6.4.2.1 Optimisation Results

It was observed that both MDF and CO optimisation frameworks could not converge to an optimum solution using the *fmincon* optimisation algorithm as the system level optimiser. However, the ATC framework converged to an optimum solution after 11 iterations, in 9039 seconds. For the ATC implementation, the termination constraints ε_1 and ε_2 (see Equation 6.14) were set as $\varepsilon_1 = \varepsilon_2 = 0.01$.

The ATC solution for the diesel engine case study is illustrated in Table 6.23. This table summarises the optimum setting for all the 58 calibration parameters, and the corresponding calibration objectives, i.e. FC (kg/hr) and Pm (gr/hr), and NOx (gr/hr), for each engine steady state minimap point. Also, the total FC, Pm and NOx responses over the NEDC drive cycle are shown in this Table 6.23.

Minimap	AFS (mg/str)	BP (hpa)	LP (%)	MIT (BTDC)	IQ (mg/inj)	FRP (hpa)	FC (kg/hr)	Pm (gr/hr)	NOx (gr/hr)
1	284.9	1098.3	85.5	-2.4	2.0	6.42E+05	0.32	0.12	0.22
2	234.0	1045.2	54.2	-1.1	NA	6.52E+05	0.28	0.09	0.34
3	286.4	1105.8	83.6	0.7	2.0	6.47E+05	0.52	0.10	0.40
4	500.6	1397.5	86.5	-0.5	NA	1.00E+06	0.39	0.37	1.28
5	223.3	NA	12.6	-4.4	NA	6.54E+05	0.07	0.01	0.35
6	317.7	1028.7	95.6	0.4	1.1	7.30E+05	0.20	0.01	0.49
7	441.2	1285.8	89.9	-3.8	1.7	7.00E+05	0.51	0.45	0.98
8	623.1	1667.5	91.6	0.9	NA	1.04E+06	0.08	0.04	0.44
9	414.3	1272.7	85.7	-0.1	NA	9.96E+05	0.21	0.17	0.41
10	519.7	1557.4	87.0	-1.6	NA	8.64E+05	0.49	0.48	0.83
11	633.4	1703.7	90.0	-0.8	NA	1.18E+06	0.34	0.16	1.13
						Overall	3.42	2.00	6.87

Table 6.23: ATC solution for the diesel engine case study

Overall, the ATC solution outperformed the benchmark solution (see Table 4.12) delivering an improvement in fuel consumption of **2.5%** over the NEDC drive cycle. Figure 6.34 illustrates the fuel consumption improvement achieved by the ATC solution for each minimap point, and the improvement on the total fuel consumption over the drive cycle, Equation 6.22.

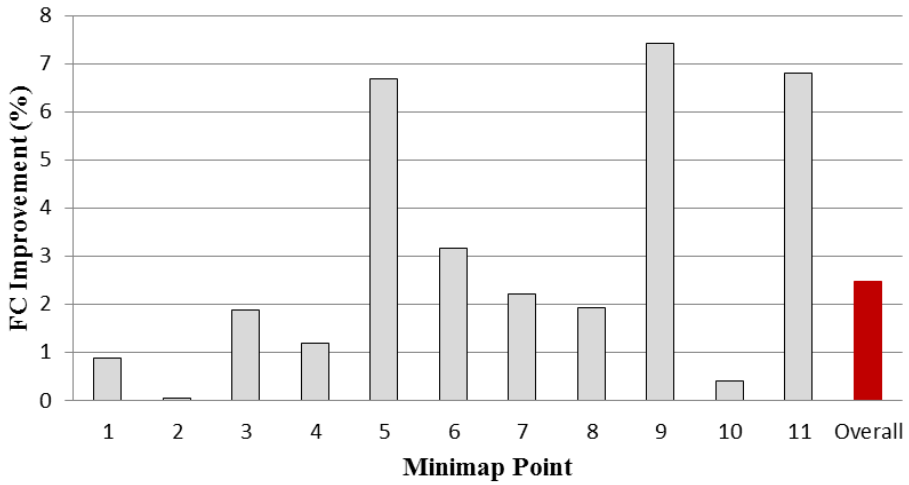


Figure 6.34: Fuel consumption (FC) improvement over the benchmark solution for the reference drive cycle (NEDC)

Figure 6.35 to Figure 6.39 illustrate the optimal solutions for each actuator (i.e. AFS, BP, LP, MIT, and FRP) across the 11 minimap points as a parallel axis plot, comparatively for ATC Versus the benchmark solution. For convenience, the graphical representation shows calibration parameters converted to coded units, between [0 1]. Note that the BP (boost pressure) was not used as a local

calibration variable for minimap point 5 (shown as N5), therefore the solution appears as zero in the corresponding graph in Figure 6.36. It is seen that the while the ATC solutions are overall in the same range as the benchmark calibration solutions, the actuator map smoothness is better for most actuators.

Figure 6.40 illustrates the overall Pm and NOx responses (i.e. over the NEDC drive cycle) for ATC solution comparatively to the benchmark, showing that the ATC solution is within the global engineering constraints limits set for the hot calibration test (see Table 4.10).

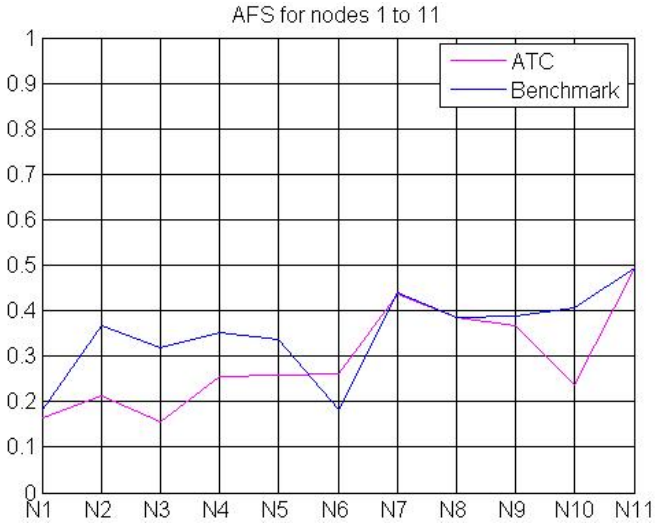


Figure 6.35: Air Mass Flow (AFS) optimal solutions

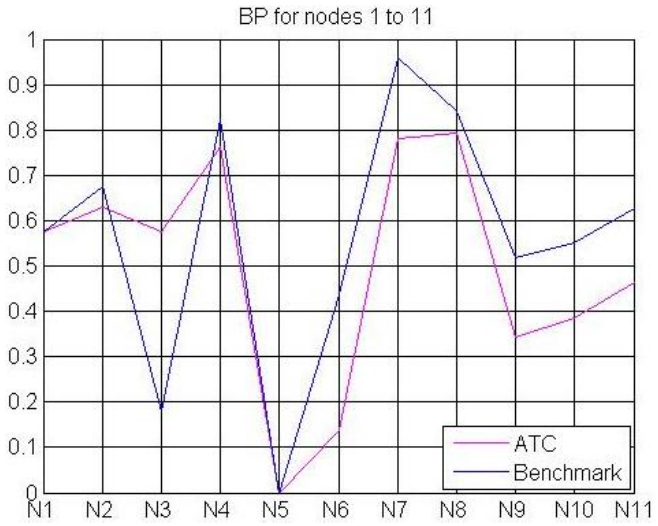


Figure 6.36: Boost Pressure (BP) optimal solutions

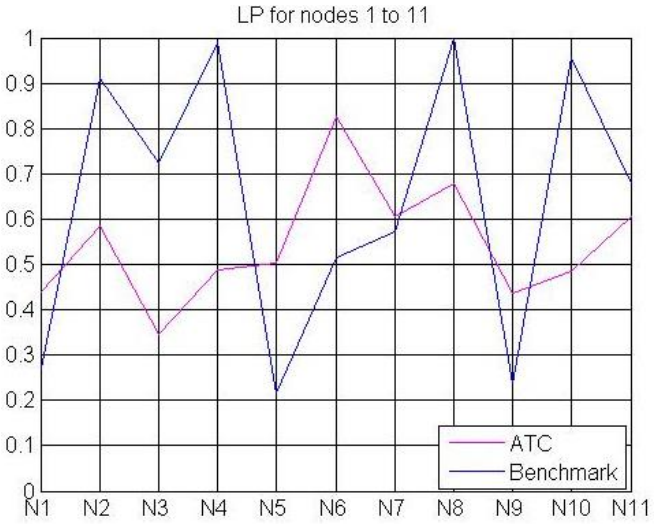


Figure 6.37: Low Pressure EGR fraction (LP) optimal solutions

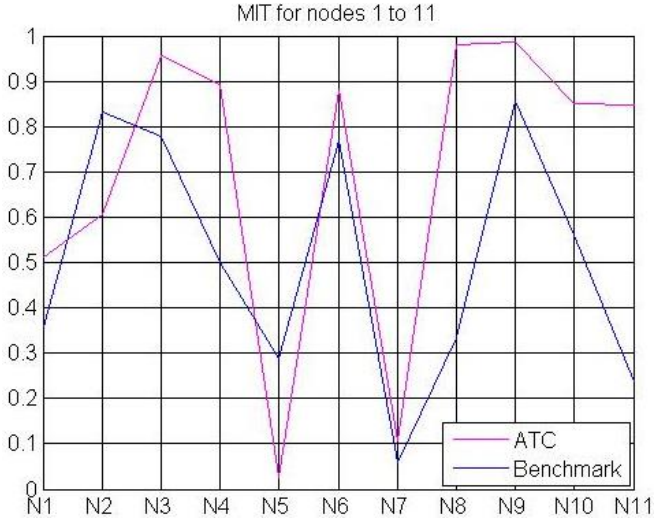


Figure 6.38: Main Injection Timing (MIT) optimal solutions

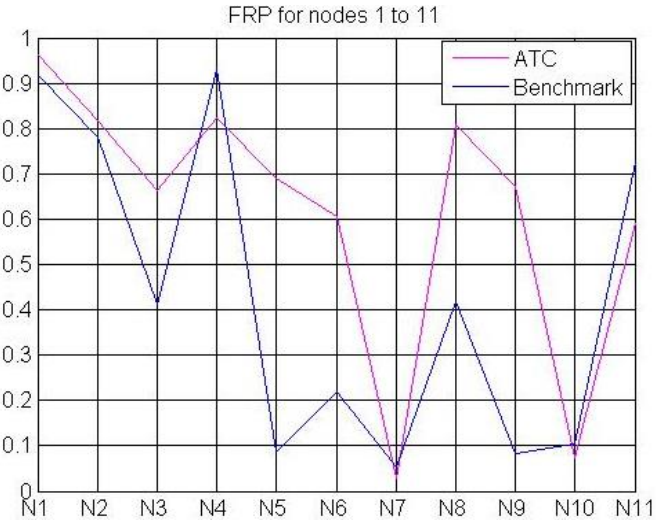


Figure 6.39: Fuel Rail Pressure (FRP) optimal solutions

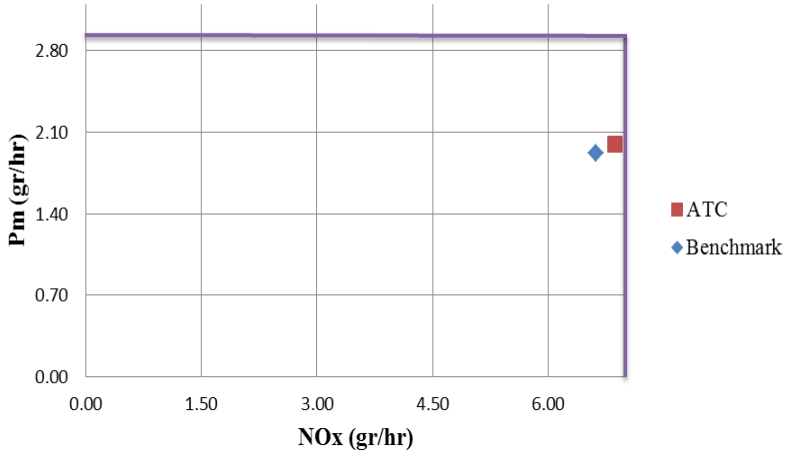


Figure 6.40: NOx vs. Pm, comparison of the optimisation solutions within the global limits

Figure 6.41 illustrates the combustion noise at each minimap point, against the noise upper limits E_i (see Table 4.11). This figure demonstrates that the ATC solution could satisfy the local constraints on the noise response at all the minimap points. Also, Figure 6.42 illustrates the distance between the ATC optimum solution and the boundary limits at each minimap point. In this figure, a negative distance value indicates that the solution is within the convex hull, thus; the ATC optimum solution at all the minimap points are within the convex hull (i.e. design boundary limits).

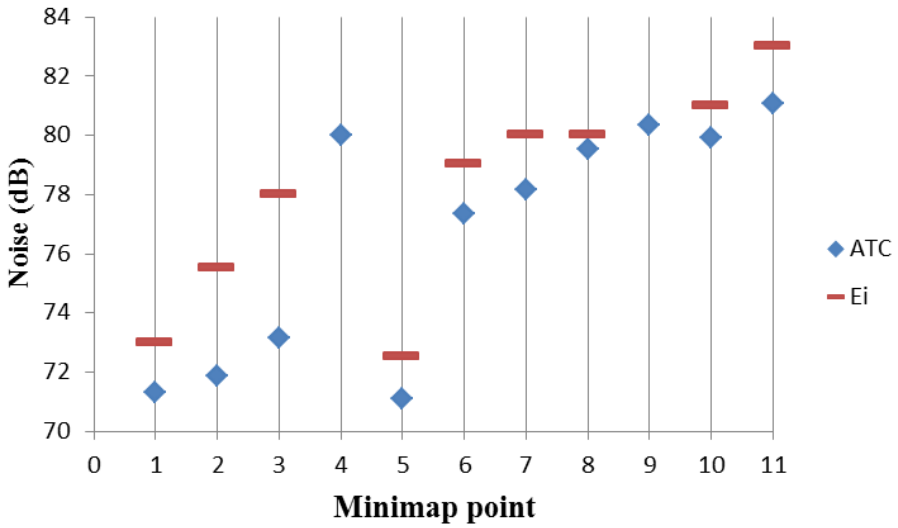


Figure 6.41: Noise generated by the CO solution, against the Noise upper limits (E_i)

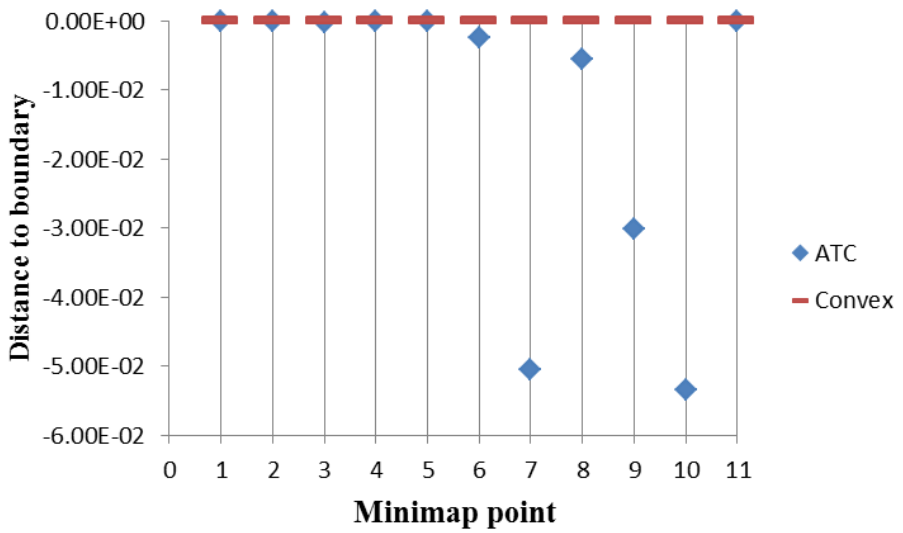
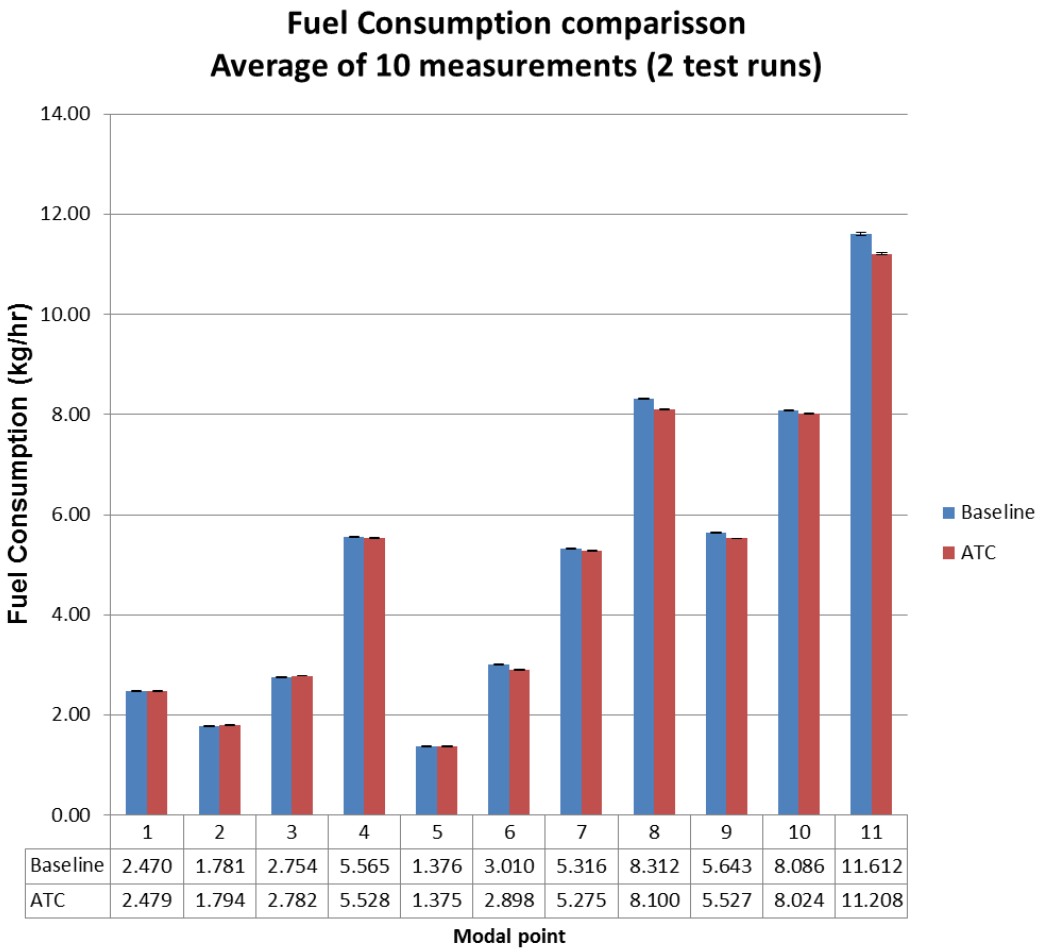


Figure 6.42: Distance to the convex hull boundary limits for the CO solution

6.4.2.2 Validation Test

The calibration settings from the ATC optimisation process were run on a test engine at Jaguar Land Rover testing facility to verify whether the ATC solution gave a significant fuel consumption benefit at the minimap test points. A back to back test was used where the baseline calibration settings were first applied, followed by the ATC generated settings. This test was repeated 10 times to gain statistical confidence in the result. The fuel consumption for the ATC solution and the benchmark setting was then calculated by averaging of all 10 test runs at each minimap point, as illustrated in Figure 6.43. This figure demonstrates that the ATC solution reduced the fuel consumption at most of the minimap points, compared to the benchmark solution. The validation test results showed a **0.8%** improvement in overall fuel consumption from the ATC optimum settings.



	1	2	3	4	5	6	7	8	9	10	11
Baseline	2.470	1.781	2.754	5.565	1.376	3.010	5.316	8.312	5.643	8.086	11.612
ATC	2.479	1.794	2.782	5.528	1.375	2.898	5.275	8.100	5.527	8.024	11.208

Figure 6.43: Validation test: the ATC solution against the benchmark setting, conducted at the Jaguar Land Rover testing facility

6.4.3 Discussion of MDO Results for the Diesel Engine Case Study

The main scope of conducting the diesel engine case study was to validate the robustness of the developed MDO frameworks on a higher dimensional steady state engine calibration optimisation problem. It was observed that MDF and CO approaches could not deliver a feasible solution for this case study using the *fmincon* optimisation algorithm as the optimiser, however; the ATC framework converged to an optimum solution after 9039 seconds. One of the main reasons for robustness of the developed ATC framework in handling highly constrained optimisation problems is its strength in ensuring a convergent coordination strategy, using the augmented Lagrangian penalty function [206]. For the diesel engine case study, the ATC framework promoted a more extensive and effective search of the design space compared to the single-objective optimisation approach which was used to generate the

benchmark calibration solution (i.e. discussed in Chapter 4). This is demonstrated by the significant improvement (2.5%) achieved for the fuel consumption (FC) over the target drive cycle, which was set as the main objective at the system level.

It should be highlighted that although CO and MDF approaches could not deliver a feasible solution, it doesn't mean that they cannot be employed for the complex steady state calibration optimisation problems. While many different optimisation frameworks can be used to formulate a given optimisation problem, and just as many optimisation algorithms may be used to solve a given optimisation problem [27]. In other words, both the choice of optimisation algorithm to solve an optimisation problem, and the choice of optimisation architecture to formulate an optimisation problem have a significant influence on the solution quality and the computational efficiency. For instance, applying a global optimisation algorithm (e.g. GA algorithm or PSO) instead of the applied gradient based optimisation algorithm (i.e. *fmincon*) would improve the performance of the developed MDF and CO approaches [27]. However, applying a global optimisation algorithm instead of a gradient based algorithm may inherently attract computation time penalty [140], [149].

As an example, Matlab Hybrid Genetic algorithm, which is available in Matlab optimisation toolbox [228], was used as the system level optimiser to solve the developed MDF and CO frameworks for the diesel engine case study. Matlab Hybrid GA combines the GA algorithm with a Sequential Quadratic Programming (SQP) algorithm, to increase the speed of the convergence. The Hybrid GA algorithm was run while the 'Population Size' was 500, 'Crossover Rate' was 0.8, 'Mutation Rate' was 0.05, 'Elite Size' was 10, and the *fmincon* algorithm was set as the SQP algorithm. Also, since the GA algorithm is a stochastic process and might converge to different optimum solutions, each of the MDF and CO frameworks was run 5 times. It was observed that using the Hybrid GA optimisation algorithm could not help the MDF framework to converge into a feasible solution, however; the CO framework converged to an optimum solution after 5 iterations of GA algorithm followed by 8 *fmincon* iterations, within 117833 seconds (around 33 hours).

The optimum calibration solution for the diesel engine case study using the CO optimisation framework is illustrated in Table 6.24. This table summarises the optimum setting for all the 58 calibration parameters, and the corresponding calibration objectives, i.e. FC (kg/hr), Pm (gr/hr), and NOx (gr/hr), for each engine minimap point. Furthermore, the total values of calibration objectives over the NEDC drive cycle are also shown in this table.

Minimap	AFS (mg/str)	BP (hpa)	LP (%)	MIT (BTDC)	IQ (mg/inj)	FRP (hpa)	FC (kg/hr)	Pm (gr/hr)	NOx (gr/hr)
1	289.9	1100.8	89.2	-2.8	2.0	5.41E+05	0.34	0.11	0.24
2	234.8	1018.0	52.6	-2.4	NA	5.87E+05	0.28	0.05	0.38
3	296.7	1106.8	84.7	-0.1	2.0	6.00E+05	0.52	0.13	0.50
4	506.7	1370.6	90.1	-2.6	NA	9.31E+05	0.40	0.39	1.36
5	238.2	NA	20.1	-4.4	NA	6.11E+05	0.07	0.01	0.40
6	328.1	1100.7	94.6	0.9	2.5	6.99E+05	0.20	0.01	0.42
7	409.9	1240.8	95.0	-4.0	2.3	8.84E+05	0.52	0.73	0.66
8	616.0	1594.8	93.5	0.3	NA	9.53E+05	0.08	0.06	0.46
9	425.2	1311.3	91.4	-0.4	NA	9.85E+05	0.22	0.13	0.40
10	548.9	1646.8	85.2	-2.0	NA	9.89E+05	0.50	0.27	1.03
11	639.2	1720.4	89.3	-0.7	NA	1.13E+06	0.34	0.22	1.10
						Overall	3.47	2.11	6.90

Table 6.24: Optimum setting for the calibration parameters using CO framework

Overall, the CO solution outperforms the benchmark calibration setting (see Table 4.12) delivering an improvement in fuel consumption of **1.16%** over the NEDC drive cycle, however; it is worse than the ATC optimum solution (i.e. Table 6.23).

Table 6.24 shows that the overall Pm and NOx could meet the global constraints for each of these responses over the NEDC drive cycle (see Table 4.10). Moreover, Figure 6.44 and Figure 6.45 show that the CO optimum solution could fulfil the local constraints set for the diesel engine case study. Figure 6.44 shows that the combustion noise values are less than the upper limits at each minimap point (see Table 4.11). Also, Figure 6.45 illustrates the distance between the CO optimum solution and the boundary limits at each minimap point. Given that a negative distance value in this graph confirms that the corresponding solution is within the convex hull, Figure 6.45 demonstrates that the CO optimum solution is within the design boundary limits at all the minimap points.

It is worth mentioning, given that MDF architecture handles all the system constraints in a single level, that this framework might require a much larger population size (e.g. 20,000 to 100,000 [254]) for the GA algorithm to avoid not finding a feasible solution.

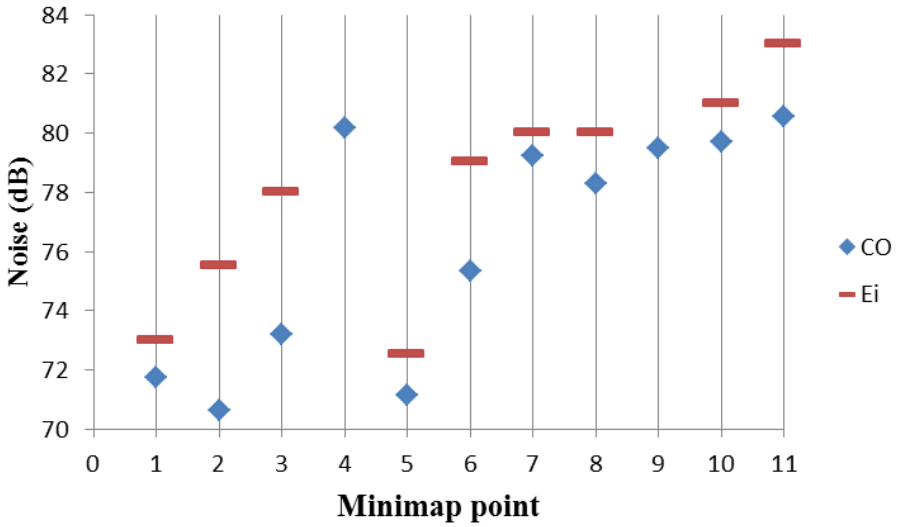


Figure 6.44: Noise generated by the CO solution, against the Noise upper limits (E_i)

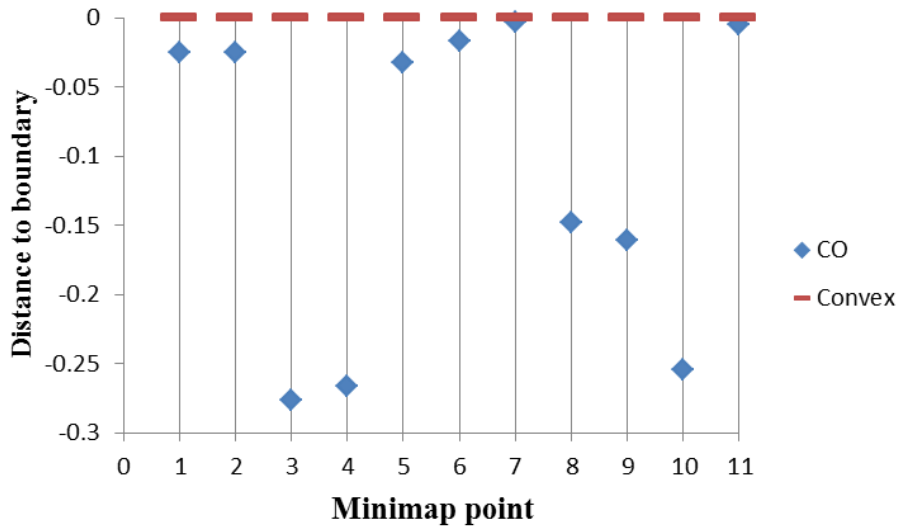


Figure 6.45: Distance to the convex hull boundary limits for the CO solution

6.5 Summary, Discussion and Conclusions

In this chapter, the conventional 2-stage calibration optimisation approach (i.e. consists of 'Local' and 'Global' optimisation levels) was implemented for the GDI engine case study over 2 calibration cases ('Case 1': over the single injection models, and 'Case2': over the double injection models). In this research, the NSGA II evolutionary optimisation algorithm was employed to conduct the local level optimisation (i.e. studying the trade-off between the FC and Pn responses). Also, to define the global optimum solution, all the possible combinations of the local solutions were exhaustively evaluated.

Subsequently, the shortcomings of the 2-stage optimisation method were studied to re-analyse the steady state calibration optimisation problem and formulate it using MDO frameworks. In this chapter, 3 MDO strategies were implemented for the GDI case study calibration optimisation problem: Multidisciplinary Feasible (MDF), Collaborative Optimisation (CO), and Analytical Target cascading (ATC). Then, results from the implementation of the MDO frameworks were presented comparatively with the baseline calibration solution, followed by a discussion of the results as shown in Table 6.17 and Table 6.18. Furthermore, the developed MDO frameworks were similarly applied for a complex steady state engine calibration optimisation problem based on a diesel engine case study.

Overall, given the structure of the steady state calibration problem, which involves at least 2 hierarchical levels, the MDO approaches offer a natural framework for optimisation problem formulation. This research has demonstrated that the MDO frameworks (particularly ATC framework) can deliver strong benefits for the steady state engine calibration optimisation problems. The advantages of the MDO frameworks can be summarised as follows:

- The GDI engine case study analysis has demonstrated that the MDO frameworks outperform the 2-stage calibration approach in terms of performance / quality for the overall (over the drive cycle) calibration results. Thus, the MDO frameworks addressed the weakness of the 2-stage process that it is not 'goal' focused on the global calibration objective.

- The MDO frameworks allow for calibration engineering preferences to be included in the optimisation problem formulation, removing the need for calibrator input in the optimisation process. This has been illustrated in the GDI calibration case study by incorporating the calibration preference for a ‘smooth actuator map’ through formulation of a constraint on the maximum actuator change, i.e. to be less than half of the design space for each calibration parameter.

- The MDO framework can be flexibly extended to accommodate problems with large dimensionality, with a large number of variables and constraints. This has also been illustrated by easily adapting the developed MDO frameworks for the GDI engine case study into a larger framework to handle the diesel engine case study.

In conclusion, it was seen that for both the GDI and Diesel engine case studies the ATC framework outperformed the other MDO approaches in solution quality, particularly for the diesel engine case study that the optimisation problem was more complex. In ATC formulation, the use of the augmented Lagrangian penalty function to co-ordinate the convergence across the hierarchy ensures a relatively fast and robust convergence compared to the other MDO frameworks discussed in this thesis. Therefore, it can be argued that the ATC architecture should be the MDO framework of choice to solve the complex steady state calibration optimisation problems.

Chapter 7: Discussion, Conclusions and Future Work

7.1 Discussion

The main aim of this thesis was to improve the conventional steady state engine mapping and calibration process and validate it through case studies. The adopted engine mapping and calibration process in this research was based on the steady state model based calibration process, which is a common practice in automotive industry [1]–[8].

Accordingly, the objectives of the research were to, firstly, enhance the steady state engine mapping process by developing a novel DoE strategy to plan the steady state experiments at each engine minimap point, and then, to improve the steady state engine calibration optimisation process by applying flexible multidisciplinary optimisation frameworks to formulate these complex optimisation problems.

The developed DoE strategy and MDO optimisation frameworks were studied in the context of the steady state mapping and calibration of the AJ133 GDI engine case study. The main objective of the GDI engine case study was to minimise the fuel consumption over the NEDC drive cycle aiming to satisfy the new European emission legislation (i.e. EURO VI) based on particulate numbers (Pn). The steady state data collection process was conducted at the university of Bradford testing facility, as discussed in Chapter 4.

Furthermore, the developed MDO approaches were validated based on a diesel engine case study. The main objective of the diesel engine case study was to minimise the fuel consumption over the NEDC drive cycle with respect to the new European emission legislation (i.e. EURO VI) for Pm and NOx. For this case study the required engine models and benchmark calibration solution were provided by the sponsoring company.

This chapter presents a critical review of the strategy taken to conduct this research, including the key developments and results in both *steady state engine mapping* and *steady state calibration optimisation* area.

7.1.1 Development of a Sequential DoE Strategy for Engine Mapping Experiments

This research introduced a novel exploration based sequential DoE strategy based on a Model Building - Model Validation sequence (i.e. MB-MV DoE strategy) to improve the shortcomings of the conventional DoE methods commonly used in practice, typically based on a large “one-shot” DoE [22], [24], [58], [73]–[75], [86], [136], [220]. The necessity of designing a new DoE strategy was based on the need to efficiently plan a DoE for an unknown engine mapping application, where the behaviour of engine responses over the range of calibration parameters and the total number of required test points to deliver highly accurate response models are not known as a priori.

The MB-MV strategy is based on sequential modification of an initial small size OLH DoE (i.e. MB design) with subsequent OLH designs of smaller size (i.e. MV designs) to achieve response models of target accuracy. In the proposed DoE strategy, while each MB and MV are individually OLH DoEs, the joined MB + MV set is no longer a Latin hypercube DoE. This departure from the strict OLH rule offers the sequential DoE framework the flexibility needed to adapt to practical problems, where the number of iterations in the sequence cannot be known in advance.

The main feature of the sequential MB-MV DoE framework is that by monitoring the fitted models accuracy (i.e. using several statistical criteria such as PRESS RMSE [54], Validation RMSE [56], and Relative Error [55]) testing can be stopped when the models are sufficiently accurate, thus reducing unnecessary further testing which will add little additional information (i.e. avoid over-sampling). Conversely, a further MV OLH DoE can be added if any of the response models is not sufficiently accurate for prediction purposes (i.e. avoid under-sampling). So, monitoring the model accuracy addresses the limitation of one-shot DoE strategies (i.e. Classical [14], [60], Optimal [60], [72], and Space filling [9]–[13] DoEs), which require defining the total number of test points in advance.

Some of the benefits of using space filling OLH designs as the batches of the test points are:

- MB-MV DoE strategy covers the whole range of the parameters at each sequence of the DoE modification (i.e. good projection property), which consequently enhances the exploration process compared to the available sequential space filling DoE approaches in literature [65], [100], [108], [109];
- MB-MV DoE is a fine-grained DoE strategy, i.e. improved sequentially with small batches of MV DoEs, compared to the similar LH based sequential DoEs such as Quasi-LHD [119]. This feature is an important requirement for a sequential DoE strategy in order to prevent over-sampling;
- No prior knowledge regarding the model type is required [9], [11], [12] (i.e. unlike the Classical and Optimal DoE strategies);
- Removing the infeasible DoE points does not defect the integrity of the design [76].

The MB-MV DoE was implemented through a custom developed PermGA algorithm, as a discrete optimisation algorithm, and using the AELH function (i.e. based on potential energy) as the objective function / space fillingness metric. The AELH metric has been justified after a critical review of the current state in literature [11], [13], on the basis that it performs better in many-dimensional situations compared to the more conventional space fillingness metrics, such as Maximin [77]–[81], and $\emptyset_p$ [80] (see Table 2.2). Similarly, the PermGA algorithm has been justified to generate uniformly distributed OLH designs, i.e. with a good space filling property, regardless of the problem dimensionality [10], [11]. The AELH fitness function was modified to adjust the algorithm to generate MV OLH DoEs with respect to the existing MB test points (see Equation 5.2). The algorithm was further adapted by introducing a penalty function (see Equation 5.3) to enable generation of valid DoE plans for situations where the design space is non-orthogonal, e.g. as a result of imposing linear or nonlinear constraints.

The effectiveness of the algorithm and the DoE methodology as a whole (of generating a flexible sequence based on augmenting an initial DoE) has been proved on a theoretical case study (i.e. SHCB function [229]) as well as the GDI engine mapping experiments.

Based on these case studies, it can be concluded that the algorithm is efficient and robust in generating uniform space filling DoE sequences for practical many-dimensional problems, which have been recognised in literature as a significant challenge [11], [13]. As it has been shown in Figure 5.15, Figure 5.36 and Figure 5.62, the developed PermGA using the AELH function could generate uniform space filling designs regardless of the problem dimensionality. These figures also illustrate that the variability of Euclidean distances between the test points (i.e. as a metric for uniform distribution of test points within the design space) is enhanced by adding the subsequent DoEs, which indicate the capability of modified AELH function to generate uniformly distributed DoE points even after a number of MV sequences. For all the case studies, it was seen that the generated individual DoEs at each sequence, and over the whole DoE process are quasi-orthogonal, where the correlation (*r*) between any of the two calibration parameters was always acceptable ($-0.3 \leq r \leq 0.3$) [90].

Furthermore, the effectiveness of the proposed MB-MV DoE strategy has been illustrated in Figure 5.17, Figure 5.40 and Figure 5.41. These figures show that the accuracy of the fitted models can be monitored (i.e. using statistical metrics) after each sequence of MB-MV process, which facilitates the calibration engineer to stop the data collection process when the target model accuracy is achieved. For instance, it was shown that the target model accuracy for the GDI engine was achieved by using fewer test points than the common practice in industry, for both injection cases, as summarised in Table 5.5 and Table 5.8. Also, the MB-MV DoE strategy addresses the shortcoming of 'one-shot' space filling DoEs to predict the response behaviour around the boundary limits [156], by sequentially collecting more data from these area, as shown in Figure 5.18.

The main advantages of the MB-MV DoE strategy for the steady state engine mapping problems, compared to the other sequential approaches which have been used in this area (such as Sobol sequence [106], and D-Optimal design augmentation DoE which is available in CAMEO toolbox [226]) are:

- MB-MV DoE strategy generates the new set of MV points considering the location of existing MB test points, while generation of new test points using Sobol sequence is quasi-random. Therefore, it is expected that MB-

MV DoE strategy enhances the exploration process since it preserves a better space filling property.

- MB-MV DoE strategy does not require pre-definition of the model type, however; the sequential DoE approach in CAMEO is based on an 'Optimal' design, which requires determination of model type in advance.
- MB-MV DoE strategy integrates well with the steady state engine mapping process in practice [7] (i.e. which consists of 'screening', 'model building', and 'model validation' stages).
- Studying the applicability of using an OLH DoE for the purpose of screening, it can be argued that OLH DoEs can be used as an attractive alternative to the CAMEO on-line screening strategy. The CAMEO screening strategy is based on searching the feasible design space for each parameter by an on-line centre-out unidirectional "excursions" [241], [242]. While, a space filling OLH DoE not only considers the whole range of each parameter to determine the unstable area, but also considers the possible combinations of parameters (such as overlap which is dependent on IVO and EVC variables) to further narrow down the feasible design space.
- By applying the MB-MV DoE strategy, the screening DoE can be used as the first stage of the 'MB-MV' process, as applied for the GDI engine case study. This strategy not only guarantees a good exploration of design space through application of a space filling DoE strategy (i.e. although size of the DoE sample is also important), but also enhances the overall steady state testing efficiency by re-using the feasible screening DoE test points in the mapping stage. Moreover, since the design space regions of interest can be reduced over subsequent MV sequences, this DoE strategy would consequently reduce the total number of required test points by avoiding sampling from the unimportant parts of design space (i.e. from calibration's points of view).
- MB-MV DoE strategy performs well with complex statistical modelling techniques (e.g. RBF models with different Kernel functions), which is due to ability of these modelling techniques to confront the noisy data (i.e.

which is the case for engine data) with a good generalization capability [123].

The main concern with using a sequential DoE strategy for engine experiments is that test cell variability (e.g. atmospheric pressure, cell temperature or humidity) could significantly influence the quality of the data, and by implication the quality of the models fitted. Therefore, the practical implementation of this methodology for engine mapping and calibration studies substantially relies on the stable operation of the engine test facility, to ensure that the effect of other sources of variation in the engine testing facility do not affect the stability of the test data collected. In order to address this issue, and given that test automation methods have progressed significantly over the recent years which enable testing to be conducted much faster, the common industrial approach is to collect all data in one go by conducting larger 'one-shot' experiments. It is however often the case that models of adequate quality cannot be fitted for all engine responses of interest, which indicates that further data (and further DoE experiments) are needed.

The developed DoE methodology and algorithm could be applied to other disciplinary areas, in particular for CAE based experiments (where the MB-MV method originates). However, this would require further validation work of the methodology for this area, in particular on the following aspects:

- Problem dimensionality: since CAE problems often deal with large dimensional design spaces, the performance of the algorithm with an increasing number of variables should be studied in relation to the ability of generating uniform MB-MV sequences of different sizes;
- Non-orthogonality of the design space: the ability and performance of the PermGA algorithm to deal with complex irregular design spaces requires further testing and validation;
- Computational performance of the algorithm: such as studying the effects of PermGA tuning-up parameters (i.e. population size, mutation rate, crossover rate) which can substantially influence the performance of GA optimisation algorithm;

- Criteria for design augmentation through infill points: this work has only considered uniformity (ultimately expressed in terms of the Euclidian distance between test points) as criteria for generating infill points for design augmentation. In literature several other strategies have been considered [105], [108], generally based on using information from the model building stage into the planning of the next experiment (e.g. by maximising the expected improvement [17]);
- The proposed sequential DoE methodology could have a significantly enhanced effectiveness if it would be fully integrated with the engine data collection process and the test automation software. This would likely require some further enhancement of the algorithm to increase the computation speed, as well as the development (and validation) of a robust fast response modelling method (e.g. based on Gaussian Kriging, which requires less test measurements due to strong interpolation among the measured test points [22], [49], [127]). This will enable engine test data to be collected in “one go”, yet ensuring that adequate response model quality is achieved with minimum test expense. Noteworthy, given the restrictions on the test automation capability at the Powertrain Research Centre at the University of Bradford, it was not possible to evaluate and prove the effectiveness of this strategy.

While all of the above considerations could provide an enhancement for engine mapping experiments, they fell beyond the scope of current research, and provide direction for further work. The advantages and disadvantages of the proposed MB-MV DoE strategy are summarised in Table 7.1.

<table>
<tr><th>Framework</th><th colspan="2">Summary of the key developments</th></tr>
<tr><td rowspan="3">MB-MV DoE Strategy</td><td colspan="2">Developed a novel exploration based sequential DoE strategy based on OLH DoE designs:
- Developed a custom PermGA algorithm based on the AELH fitness function to generate OLH designs;
- Modified the AELH fitness function to generate infill model validation (MV) OLH designs for symmetric design spaces;
- Modified the AELH fitness function using a penalty function to generate infill MV OLH designs for asymmetric design spaces.</td></tr>
<tr><td>Advantages</td><td>Disadvantages</td></tr>
<tr><td>- Good space filling property;
- Good orthogonality property (i.e. quasi-orthogonal);
- Good projection property (covers whole range of each variable’s design space);
- A fine-grained DoE strategy;
- No Model type is required in advance;
- Total number of test points is not required in advance;
- Suitable for both symmetric and asymmetric design spaces;
- Guarantees to achieve the target model accuracy (by avoiding under-sampling);
- Potentially reduces the testing effort (by avoiding over-sampling);
- Good integration with the procedure of steady state engine mapping process (consisting of screening, model building and model validation);
- Enhances the steady state engine data collection process by defining the design space regions of interest through a screening DoE, and reusing the feasible screening test points for the model building process.</td><td>- PermGA requires tuning-up the algorithm parameters;
- It is an off-line DoE strategy;
- The practicality of this DoE strategy for steady state engine mapping applications substantially relies on the stable operation of the testing facility (since the data is collected through several independent testing efforts).</td></tr>
</table>

Table 7.1: Summary of key developments in MB-MV DoE implementation

7.1.2 Development of MDO Strategies for Engine Calibration Optimisation

This research proposed and applied the multidisciplinary optimisation frameworks as an alternative optimisation strategy to formulate the steady state engine mapping calibration optimisation problems, to address the limitations of conventional calibration optimisation strategies (e.g. conventional 2-stage approach [2], [21], [22]). The need for a new optimisation strategy has become more essential considering the ever increasing developments of engine technologies, which are consequently adding more calibration parameters to the optimisation process. Therefore, there is a need for an efficient optimisation strategy which can potentially be applied to optimise the future engine calibration problems.

The performance of the conventional 2-stage calibration optimisation approach to formulate the engine calibration optimisation problems have been investigated on the GDI engine case study, using the engine response models fitted in Chapter 5. For this research, the NSGA II evolutionary optimisation algorithm was employed to find the local trade-off between the GDI engine key responses (i.e. FC and Pn). The global solution was then determined through exhaustive evaluation of all the possible combinations of the local solutions, subject to the engineering requirement of delivering a 'smooth' map. Accordingly, the main shortcomings of the 2-stage calibration optimisation strategy can be summarised as following:

- It requires studying the local trade-offs at each minimap point, as shown in Figure 6.1;
- The whole optimisation process is time consuming and iterative. For instance, for the GDI engine problem where 4 local candidates were selected at each minimap point, there were (4^6) possible global solutions;
- The optimisation process is not goal focused. It is fundamentally base on selecting the best combination of local solutions, e.g. as summarised in Appendix II;
- This approach is not convenient for high dimensional calibration optimisation problems with many minimap points to calibrate.

Given the limitations of the 2-stage optimisation approach, particularly for high dimensional calibration problems, a common industrial approach is to use the available commercial toolboxes (such as CAMEO [226]) as an alternative optimisation method. In this research, CAMEO toolbox was applied by the calibration engineers at the sponsoring company to optimise the diesel engine calibration optimisation problem, i.e. presented in Chapter 4 as the benchmark solution (see Table 4.12). Although using a toolbox like CAMEO would enhance the coherency of searching process by formulating the optimisation problem as the 'weighted sum' of all objectives over the drive cycle, it still has to overcome the difficulty of finding a feasible solution subject to all the constraints introduced, which can be a very difficult task even for the global optimisation algorithms [43] (e.g. GA and PSO algorithms).

The steady state engine calibration optimisation problem was analysed on the GDI engine case study in Chapter 6. This analysis pointed out into 2 levels of optimisation: 'Local Level' optimisation which is associated with the objectives and constraints at each minimap point (see Equation 6.1), and 'global level' optimisation which is related to the objectives and constraints over all the minimap points / or drive cycle (see Equation 6.4).

After conducting a comprehensive literature review on the current state of optimisation strategies, it was depicted that MDO optimisation strategies [27]–[33] offer a problem simplification strategy by decomposing the high dimensional optimisation problem into a number of lower dimensionality optimisation tasks (i.e. reduce the analysis cost) and use a strategy to keep the whole optimisation process consistent [35]. Accordingly, MDO approaches were considered as an alternative optimisation strategy to formulate the steady state engine calibration optimisation problems, which can solve both levels of optimisation (i.e. both local and global optimisation levels) simultaneously.

Three MDO approaches (i.e. MDF, CO and ATC) were applied in this research to formulate the steady state engine calibration optimisation problems, which have been chosen after a critical review of the current state in literature:

1- MDF: this single-level MDO architecture [39] was applied due to its similarity to the optimisation techniques used by the current optimisation toolboxes in automotive industry (e.g. CAMEO).

2- CO: this bi-level MDO architecture [33], [186], [197] was applied since it has been frequently used in literature as a benchmark to compare performance of other MDO approaches [27], i.e. it does not involve tuning-up the framework parameters.

3- ATC: this multi-level MDO architecture was applied since it has been originally developed to handle complex optimisation problems in automotive industry [42].

Previous research was conducted by Yin [43], where he proposed and applied the MDF and CO multidisciplinary optimisation frameworks to formulate the steady state calibration optimisation of a diesel engine case study. However, to the extent of my knowledge, the ATC optimisation architecture has not ever been implemented to solve an engine calibration optimisation problem.

The problem formulation and solution quality of the applied MDO frameworks have been presented for two engine case studies:

1- The GDI engine case study, where the calibration process was conducted over two separate circumstances (‘Case 1’ with 24 calibration parameters, and ‘Case 2’ with 32 calibration parameters).

2- The diesel engine case study (i.e. 58 calibration parameters).

To formulate the chosen MDO architectures, the MDF framework was associated to the case studies by treating the engine performance over all minimap points (i.e. drive cycle) as the system level optimisation, while considering each minimap point as a subsystem analyser, as shown in Table 6.5 and Table 6.20. The CO and ATC optimisation frameworks were associated to the engine case studies by treating the engine performance over all minimap points as the system level optimisation, while considering each minimap point as a discipline at the subsystem level optimisation, as illustrated in Figure 6.9 and Figure 6.13 for the GDI engine, and Figure 6.22 and Figure 6.23 for the diesel engine. A unique feature of engine calibration optimisation problems for

both GDI and diesel engine case studies was that there was no coupling between disciplines (i.e. no variables are shared among the disciplines), thus, it was assumed that the disciplines are working independently where the variables can be treated as ‘pseudo-independent’.

The main difference between ATC and CO implementations is the way whole system consistency is maintained [42], [202]. The CO approach keeps the whole system consistency by an equality constraint at the system level, which is a function of discrepancy between the system level and subsystem level solutions, whereas the ATC maintains the system consistency using a penalty function. The augmented Lagrangian penalty function was justified for this research after reviewing the literature [206], which has been proved to enhance the ATC approach robustness by updating the penalty weights iteratively until the desired consistency is achieved.

The quality of MDO solutions attained for the GDI engine case study has been compared together against a benchmark solution (i.e. the current calibration setting on engine set by the sponsoring company, shown in Table 4.5), as summarised in Table 6.17 and Table 6.18. Noteworthy, a gradient-based sequential quadratic programming algorithm (i.e. *fmincon* [248]) was used as the system lever optimiser to implement the MDO frameworks (i.e. instead of a global optimisation algorithm) in order to enhance the computation speed. The main findings of this research are:

- All MDO solutions could outperform the benchmark calibration solution.
- ATC delivered the best calibration solution (in terms of minimising the overall FC) compared to the other MDO approaches applied.
- The advantage of applying the ATC framework compared to the other MDO approaches was clearer for calibration ‘case 2’, which was a higher dimensional optimisation problem.
- Both MDF and CO optimisation approaches delivered advantages in terms of main calibration objectives, however; the MDF approach converged to an optimum solution faster than the other MDO frameworks. A reason for

faster convergence of MDF was the capability of this approach to handle both local and global optimisation levels simultaneously [197].

- CO resulted in an exhaustive computational process in comparison with the other MDO approaches. A reason for slow convergence of the gradient-based optimisation algorithm could be the failure of this algorithm to satisfy the equality constraints [200].

Although application of MDO approaches for the GDI engine case study showed significant improvements in both overall FC and Pn responses based on the engine models, the MDO solutions have not been validated on the test bed due to time limits of the CREO project.

The applicability of applying MDO frameworks to solve the steady state engine calibration optimisation problems were also investigated on the diesel engine case study, which is a heavily constrained optimisation problem (i.e. the probability (P_f) of finding a feasible solution is 0.016 %, as discussed in Section 6.4). The benchmark solution for this case study was the current calibration setting on the diesel engine, which was defined by the calibration engineers at the sponsoring company using the CAMEO optimisation toolbox (see Table 4.12). Accordingly, the main results of this research can be summarised as:

- Using *fmincon* optimisation algorithm as the MDO's system level optimiser, only ATC could converge to an optimum solution (after 9039 seconds), as summarised in Table 6.23. The main reason for high robustness of ATC framework in dealing with heavily constrained optimisation problem is the strength of applied strategy (i.e. augmented Lagrangian penalty function) to coordinate the convergence across the optimisation hierarchy [206].
- ATC framework outperformed the benchmark solution, delivering a significant improvement in overall fuel consumption (i.e. 2.5 %).
- MDF architecture could not find a feasible solution when using the *fmincon* optimisation algorithm as the system level optimiser. A reason for low robustness of MDF approach to find a feasible solution for such a heavily constrained optimisation problems, while using a gradient-based

optimisation algorithm as the system optimiser, is that MDF can fail when the optimiser does not converge in an iteration [34].

- MDF architecture could not also find a solution when using Matlab Hybrid Genetic algorithm as the system optimiser (population size was set to 500). However, given that MDF architecture handles all the system constraints in a single level, a possible solution to this difficulty could be to run the GA algorithm with a much larger population size (e.g. 20,000 to 100,000 [254]).
- CO architecture could not find a feasible solution when using the *fmincon* optimisation algorithm as the system level optimiser. The main reason for difficulty of CO framework was the high number of equality constraints to satisfy, which could have led into failure of applied gradient-based optimisation algorithm to deliver feasible solutions at subproblems [34], [198].
- CO architecture could find a solution when using Matlab Hybrid Genetic algorithm as the system optimiser (population size was set to 500) within 117833 seconds, as summarised in Table 6.24. The attained CO solution outperformed the benchmark calibration setting, i.e. enhancing the overall fuel consumption by 1.16%. The CO architecture converged successfully to a feasible solution (unlike the MDF with Hybrid GA algorithm) since CO relaxes the optimisation problem dimensionality, by decomposing it into a number of optimisation problems of smaller size.
- The achieved solution for the CO approach is worse than ATC solution. Also, given that a global optimisation algorithm was used to solve the CO framework, the CO was computationally more expensive than ATC.
- The ATC solution was validated on a test engine at Jaguar Land Rover testing facility. The validation test showed a 0.8% improvement in overall fuel consumption compared to the benchmark solution, as shown in Figure 6.43. The main reasons for the difference between the improvements achieved from the engine models and validation test are: the diesel engine tested for validation was not the same engine tested to collect the DoE test points, and also the validation test was conducted several months after the

original DoE tests. However, considering that the ATC solution was tested for 10 times and the average was used for the comparison, it can be argued that the optimisation results are promising.

Therefore, the main advantages of MDO frameworks to formulate the steady state engine calibration optimisation problems can be summarised as:

- There is no need to analyse the local trade-offs;
- MDO architectures provide an opportunity to formulate the steady state calibration optimisation problems in a way coherent with the 2-level structure of model based calibration problems;
- MDO frameworks allow the calibration engineering preferences (e.g. ‘smooth actuator map’) to be formulated in the optimisation problem (i.e. as an objective or a constraint);
- MDO frameworks can be flexibly extended to accommodate engine calibration problems with larger dimensionality.

The developed MDO frameworks could be applied for other steady state engine calibration optimisation problems regardless of the problem dimensionality. However, there are several aspects that would require further validation work, such as following:

- Optimisation algorithms: since the real-life calibration optimisation problems are dealing with heavily constrained large dimensional optimisation problems, the performance of different optimisation algorithms (i.e. GA and PSO) should be studied in relation to the MDO approaches and as well as problem dimensionality;
- Problem formulation: the efficiency of MDO approaches can be studied in relation to different problem formulations (e.g. relaxation of problem constraints, by formulating them as a penalty term in the objective function);
- Further validation of MDO approaches: in terms of convergence properties and solution quality, on different real world calibration problems.

The above considerations fell beyond the scope of the research, thus, they can be regarded as direction for future work. The advantages and disadvantages of applied MDO frameworks for the engine case studies are summarised in Table 7.2.

<table>
<tr><th>Framework</th><th colspan="2">Summary of the Applied MDO Approaches</th></tr>
<tr><td rowspan="3">MDF</td><td colspan="2">- Applied the MDF strategy for the GDI engine case study using the fmincon optimisation algorithm;
- Applied the MDF strategy for the diesel engine case study using the fmincon and Hybrid GA optimisation algorithms.</td></tr>
<tr><td>Advantages</td><td>Disadvantages</td></tr>
<tr><td>- Better problem formulation than the 2-stage calibration strategy;
- Handling both levels of optimisation concurrently;
- Fast computation.</td><td>- Not a robust approach for heavily constrained large dimensional problems;
- Not working robustly using gradient-based optimisation algorithms.</td></tr>
<tr><td rowspan="3">CO</td><td colspan="2">- Applied the CO strategy for the GDI engine case study using the fmincon optimisation algorithm;
- Applied the CO strategy for the diesel engine case study using the fmincon and Hybrid GA optimisation algorithms.</td></tr>
<tr><td>Advantages</td><td>Disadvantages</td></tr>
<tr><td>- Better problem formulation than the 2-stage calibration strategy;
- Decomposition of the problem dimensionality;
- No parameter tuning-up is required.</td><td>- Not working robustly using gradient-based optimisation algorithms;
- Computation is relatively expensive.</td></tr>
<tr><td rowspan="3">ATC</td><td colspan="2">- Applied the ATC strategy for the GDI engine case study using the fmincon optimisation algorithm;
- Applied the ATC strategy for the diesel engine case study using the fmincon optimisation algorithm.</td></tr>
<tr><td>Advantages</td><td>Disadvantages</td></tr>
<tr><td>- Better problem formulation than the 2-stage calibration strategy;
- A target based approach, i.e. facilitating a more focused decomposition strategy;
- A robust framework regardless of the problem dimensionality;
- Performed well using gradient-based optimisation algorithms;
- Relatively much faster than CO framework.</td><td>- More optimisation parameters to tune-up in advance;
- Requires definition of targets.</td></tr>
</table>

Table 7.2: Summary of MDO implementations

7.2 Conclusions

Based on the research presented in this thesis, the main conclusions can be summarised as follows:

- It was demonstrated that the OLH and MB-MV DoE algorithms developed using PermGA can generate uniform space filling quasi-orthogonal designs covering the entire design space.
- The use of an OLH DoE for the screening experiment enabled an assessment of the relationships between calibration variables and key engine responses, leading to the establishment of a reduced design space for engine mapping experiments. As shown for the GDI engine case study, the design space range for Fuel Rail Pressure (FRP) parameter was reduced from 3 – 15 (MPa) into 8 – 15 (MPa), i.e. reduced by 42%, after analysing the behaviour of FC and Pn responses at the screening stage.
- Considering the results attained from applying the sequential MB-MV strategy for a number of case studies with different dimensionalities (i.e. having 2, 4 and 6 parameters), it was seen that the proposed MB-MV DoE strategy performs robustly regardless of the problem dimensionality and variables' design space.
- Implementation of the sequential MB-MV DoE strategy can enhance the steady state data collection process for engine applications, through delivering high quality engine models fulfilling the target model accuracy using a fewer number of test points. As illustrated for the GDI engine case study, the number of required test points at different minimap points are 30 – 50 % smaller than a one-shot design of 120 – 150 test points which was originally proposed by the sponsoring company (i.e. see Table 5.5 and Table 5.8).
- This research has demonstrated that the MDO frameworks can deliver strong benefits for the steady state engine calibration optimisation problems by reducing the optimisation problem dimensionality, which in effect provides the opportunity to improve the optimisation problem formulation and computational efficiency, as demonstrated in Table 6.17 and Table 6.18 for the GDI engine case study.

- Given the structure of steady state calibration optimisation problems, which involves at least 2 hierarchical levels, MDO architectures offer a natural framework to formulate these optimisation problems.
- MDO frameworks are scalable and can be flexibly expanded to accommodate large dimensional optimisation problems, e.g. as demonstrated for the diesel engine case study with 58 parameters.
- The MDO frameworks allow the calibration engineering preferences to be included in the optimisation problem formulation, removing the need for calibrator input in the optimisation process.
- Considering the results attained from application of different MDO frameworks to the GDI and diesel engine case studies, it can be concluded that ATC architecture should be the MDO framework of choice, based on its strength in ensuring a convergent co-ordination strategy. As illustrated for the GDI engine case study, the ATC solution reduced the overall fuel consumption by 7.11 % for the single injection calibration strategy, and by 8.37 % for the double injection calibration strategy, while clearly outperformed the other applied MDO strategies for both calibration cases. Moreover, for the diesel engine case study only ATC converged to an optimum solution (i.e. using *fmincon* algorithm as the system optimiser). The ATC solution reduced the diesel engine overall fuel consumption by 2.5 %, which also underpins the robustness of ATC architecture to solve the highly complex steady state engine calibration optimisation problems.

7.3 Summary of Original Contributions

The main original contributions of this research can be summarised as:

- Development and application of a novel exploration-based sequential DoE strategy (i.e. MB-MV DoE strategy) to plan the steady state engine data collection experiments. This includes designing a DoE strategy coherent with the steady state engine mapping process, i.e. planning subsequent DoEs for ‘screening’, ‘model building’ and ‘model validation’ purposes, on the basis of OLH DoEs. Through the MB-MV DoE process the quantitative characteristics of the added blocks of DoEs (MV DoEs) are unchanged,

thus; the final DoE is not strictly an OLH design. The MB-MV DoE strategy improves the design granularity compared to the existing OLH-based sequential DoEs (e.g. sequential Nested Latin Hypercube DoE method) which in effect reduces the possibility of over – or under sampling.

- Application of MDO optimisation strategies to formulate and solve the steady state engine calibration optimisation problems. This involves analysis of the engine calibration optimisation problems, articulation of the calibration problems using MDO optimisation strategies, and validation of the developed MDO strategies on real world calibration optimisation problems. This research confirms that the applied MDO architectures (i.e. particularly ATC) have facilitated the opportunity to enhance the conventional steady state engine calibration optimisation methods by functional decomposition of the optimisation problem into a number of optimisation sub-problems. Therefore, application of MDO strategies inherently have the potential to improve the quality of optimisation solution, by providing a better opportunity to search the design space corresponding to each discipline, and also reduce the computational effort.

7.4 Future Work

Some of the future research opportunities to improve the implemented steady state engine mapping and calibration process are listed as following:

- Further development of the MB-MV DoE methodology by considering other optimal infill criteria which are directly focused on improving the model quality (rather than the uniformity of the space filling).
- Further validation of the applied penalty function to plan DoEs for asymmetric design spaces, by applying it for complex irregular design spaces. Also, boundary models can be applied to enhance generation of infill test points for highly constrained design spaces.
- Further work is needed to address the computational and statistical efficiency of the MB-MV strategy, in particular in dealing with systems of large dimensionality and non-orthogonal variable spaces, for which ensuring OLH DoE sequences with good space-fillingness and non-

collapsing properties is particularly challenging. This might also require application of different space filling metrics (such as Maximin [77]–[81], and $\emptyset_p$ [80]) to study the relation between efficiency of these space filling metrics and problem dimensionality.

- Improve the engine response modelling process by implementing Kriging models within Matlab environment.
- Integrate the MB-MV DoE strategy with a test automation software package.
- Employing an optimisation algorithm which can provide more than one optimum solution at the system level of MDO frameworks, to improve the reliability of the process (such as specious conservation algorithm [255]).
- Applying different global optimisation algorithms (e.g. GA and PSO) to study the efficiency of MDO architectures (i.e. in terms of robustness, solution quality and computation efficiency) in relation to different optimisation algorithms.
- Trying different problem formulations. For example, maximum allowable actuator change for the GDI engine case study can be formulated as a penalty function in the system level, or as an objective in the subsystem levels.
- In relation to real world calibration usage, the implemented MDO frameworks require further validation of robustness - in terms of both convergence and quality of the solutions, as well as usability by calibration engineers.
- Considering that ATC has the ability to decompose the optimisation problem into more than two levels, this optimisation framework can offer the opportunity to integrate the calibration optimisation problem with the higher levels of the systems engineering hierarchy, e.g. powertrain and vehicle system optimisation. This is important as it would enable co-development of calibration and subsystem level design; e.g. calibration optimisation could be combined with the aftertreatment and driveline system optimisation, co-ordinated by the powertrain system targets.

- Further development of ATC to have robust calibrations across multiple reference drive cycles or multiple vehicle configurations. This can be set up as a higher level objective within an ATC framework, which co-ordinates the concurrent optimisation across multiple reference drive cycles, leading to more robust calibrations.

References

[1] A. Schlober, R. Linssen, and P. Bozelie, “Model-Based Calibration of SI-Engines: Map Optimization,” in *Design of Experiments in Engine Development*, 2007, pp. 145 – 164.

[2] R. Singh, F. Campean, B. Seale, D. Grove, and M. Cary, “Evaluation of Camshaft Control Strategies using a Multi-Evolutionary Genetic Algorithm,” in *Design of Experiments in Engine Development*, 2007, pp. 165 – 183.

[3] T. Steidten, P. Adomeit, B. Kircher, and S. Wedowski, “Automated Gas Exchange Model Calibration Using an Onine Optimizer Tool,” in *Design of Experiments in Engine Development*, 2005, pp. 184 – 200.

[4] W. Lach, R., Kiehn, R., Hans, C., Siebertz, K., Platzbacker, “DoE Application within the Analytical Valve Train Development,” in *Design of Experiments in Engine Development*, 2005, pp. 221–234.

[5] J. S. Chang, J. H. Cheong, C. H. Jo, Y. H. Chi, and K. J. Yoon, “Feasibility Study on Roboust Calibratio by DoE to Minimize the Exhaust Emission Deviations from Injector Flow Rate Scatters,” in *Design of Experiments in Engine Development*, 2007, pp. 196 – 210.

[6] R. Lach, F. Weber, K. Siebertz, B. Heuvel, and E. Karvounis, “DoE Application within the Base Engine Design,” in *Design of Experiments in Engine Development*, 2007, pp. 249 – 260.

[7] K. Roepke and C. von Essen, “DoE in Engine Development,” *Qual. Reliab. Eng. Int.*, vol. 24, no. 6, pp. 643–651, 2008.

[8] T. P. Dwyer, M. R. Kianifar, W. J. Bradley, I. F. Campean, B. A. Mason, M. K. Ebrahimi, D. Richardson, and L. Beddow, “DoE Framework for GDI Engine Mapping and Calibration Optimisation for CO 2 and Particulate Number Emissions,” *Roepke Des. Exp. Engine Dev. Expert Verlag*, pp. 418–432, 2013.

[9] S. J. Bates, J. Sienz, and D. S. Langley, “Formulation of the Audze–Eglais Uniform Latin Hypercube design of experiments,” *Adv. Eng. Softw.*, vol. 34, no. 8, pp. 493–506, 2003.

[10] S. J. Bates, J. Sienz, and V. Toropov, “Formulation of the Optimal Latin Hypercube Design of Experiments Using a Permutation Genetic Algorithm,” in *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, 2004.

[11] M. Liefvendahl and R. Stocki, “A study on algorithms for optimization of Latin hypercubes,” *J. Stat. Plan. Inference*, vol. 136, no. 9, pp. 3231–3247, 2006.

[12] T. Z. Vassili Toropov Uwe, Uwe Schramm, Atul Sahai, Royston D. Jones, “Design Optimization and Stochastic Analysis based on the Moving Least Squares Method,” 2005.

[13] W. C. R. Jin, “An efficient algorithm for constructing optimal design of computer experiments,” *J. Stat. Plan. Inference*, vol. 134, pp. 268 – 287, 2005.

[14] D. M. Grove and T. P. Davis, *Engineering Quality & Experimental Design*. Essex, England: Longman Scientific & Technical, 1992.

[15] J. Seabrook, B. ROGERS, G. FARROW, and J. PATTERSON, *International Conference on Statistics and Analytical Methods in Automotive Engineering*, vol. 22. John Wiley & Sons, 2002, pp. 17–32.

[16] F. Pukelsheim, *Optimal Design of Experiments (Google eBook)*. SIAM, 2006.

[17] A. Forrester, D. A. Sobester, and A. Keane, *Engineering Design via Surrogate Modelling: A Practical Guide*. John Wiley & Sons, 2008.

[18] G. E. P. Box and N. R. Draper, *Empirical model-building and response surfaces*. Oxford, England: John Wiley & Sons, 1987.

[19] V. Toropov, “Response surface methodology,” *Online Handout, University of Bradford*. [Online]. Available: http://www.brad.ac.uk/staff/vtoropov/burgeon/thesis_luis/chapter3.pdf. [Accessed: 07-Aug-2014].

[20] T. J. Barlow, S. Latham, I. S. McCrae, and P. G. Boulter, *A reference book of driving cycles for use in the measurement of road vehicle emissions*. 2009, pp. 1–14.

[21] H. Kaji and H. Kita, “Acceleration of Experiment-Based Evolutionary Multi-objective Optimization of internal-combustion engine controllers using fitness estimation,” in *IEEE Congress on Evolutionary Computation*, 2007, pp. 1777–1784.

[22] J. Seabrook, J. Collins, and S. Edwards, “Application of advanced Modelling Techniques to the Calibration of Gasoline Engines with Direct Injection and Variable Valve Timing,” in *Design of Experiments in Engine Development*, 2005, pp. 235–245.

[23] G. R. Vossoughi and S. Rezazadeh, “Optimization of the Calibration for an Internal Combustion Engine Management System Using Multi-

Objective Genetic Algorithms," in *IEEE Congress on Evolutionary Computation*, 2005, vol. 2, pp. 1254–1261.

[24] T. Morton, R. Connors, P. Maloney, and D. Sampson, "Model-Based Optimal Calibration of a Dual Independent Variable Valve-Timing Engine," *available in Mathworks website*, 2013. .

[25] "Engine Calibration-Model-Based Calibration Toolbox-MATLAB - MathWorks United Kingdom." [Online]. Available: http://www.mathworks.co.uk/products/mbc/. [Accessed: 27-Apr-2014].

[26] R. J. Lygoe, M. Cary, and P. J. Fleming, "A Many-Objective Optimisation Decision-Making Process Applied to Automotive Diesel Engine Calibration," *Simulated Evol. Learn.*, vol. 6457, pp. 638–646, 2010.

[27] J. R. Martins and A. B. Lambe, "Multidisciplinary Design Optimization: A Survey of Architectures," *AIAA J.*, vol. 51, no. 9, pp. 2049–2075, Aug. 2013.

[28] M. Kokkolaras, R. Fellini, H. M. Kim, N. F. Michelena, and P. Y. Papalambros, "Extension of the target cascading formulation to the design of product families," *Struct. Multidiscip. Optim.*, vol. 24, pp. 293–301, 2002.

[29] H. M. Kim, D. G. Rideout, P. Y. Papalambros, and J. L. Stein, "Analytical target cascading in automotive vehicle design," *J. Mech. Des.*, vol. 125, no. 3, pp. 481–489, 2003.

[30] J. Allison, D. Walsh, M. Kokkolaras, P. Y. Papalambros, and M. Cartmell, "Analytical Target Cascading in Aircraft Design," in *Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit*, 2006.

[31] J. Allison, B. Roth, M. Kokkolaras, I. Kroo, and P. Y. Papalambros, "Aircraft Family Design Using Decomposition-Based Methods," in *Proceedings of the 11th AIAA/ISSMO MUltidisciplinary Analysis and Optimization Conference*, 2006.

[32] I. Kroo and V. Manning, "Collaborative Optimisation: Status and directions," in *8th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 2000.

[33] R. D. Braun and I. M. Kroo, "Development and Application of the Collaborative Optimization Architecture in a Multidisciplinary Design Environment," *nasa_techdocs*, vol. AIAA Paper, 1995.

[34] J. T. Allison, "Complex system optimisation: A review of Analytical Target Cascading , Collaborative Optimisation and other formulations," MSc Thesis, The University of Michigan, 2004.

[35] I. Kroo, S. Altus, R. Braun, P. Gage, and I. Sobieski, “Multidisciplinary Optimization Methods for Aircraft Preliminary Design,” NASA Langley Technical Report Server, Stanford Univ.; Stanford, CA, United States, Sep. 1994.

[36] S. Kodiyalam and J. Sobieszczanski-Sobieski, “Multidisciplinary design optimisation - some formal methods, framework requirements, and application to vehicle design,” *Int. J. Veh. Des.*, vol. 25, no. Numbers 1–2/2001, pp. 3–22, 2001.

[37] R. T. HAFTKA, “Simultaneous analysis and design,” *AIAA J.*, vol. 23, no. 7, pp. 1099–1103, Jul. 1985.

[38] S. Kodiyalam and J. Sobieszczanski-Sobieski, “Bilevel Integrated System Synthesis with Response Surfaces,” *AIAA J.*, vol. 38, no. 8, pp. 1479–1485, Aug. 2000.

[39] E. J. Cramer, J. E. Dennis, Jr., P. D. Frank, R. M. Lewis, and G. R. Shubin, “Problem Formulation for Multidisciplinary Optimization,” *SIAM J. Optim.*, vol. 4, no. 4, pp. 754–776, Nov. 1994.

[40] H. M. Kim, D. G. Rideout, P. Y. Papalambros, and J. L. Stein, “Analytical Target Cascading in Automotive Vehicle Design,” *J. Mech. Des.*, vol. 125, no. 3, p. 481, 2003.

[41] H. M. Kim, M. Kokkolara, L. S. Louca, G. J. Deladrammatikas, N. F. Michelena, Z. S. Filipi, P. Y. Papalambros, J. L. Stein, and D. N. Assanis, “Target Cascading in Vehicle Redesign: a class VI truck study,” *Int. J. Veh. Des.*, vol. 29, no. 3, pp. 199–225, 2002.

[42] H. M. Kim, T. Jiang, N. F. Michelena, and P. Y. Papalambros, “Target Cascading in Optimal System Design,” *J. Mech. Des.*, vol. 125, no. 3, pp. 474–480, 2003.

[43] X. Yin, “Application of Multidisciplinary Design Optimisation to Engine Calibration Optimisation,” PhD thesis, University of Bradford, 2012.

[44] “The CREO Project: Carbon Dioxide Reduction Through Emissions Optimisation.” [Online]. Available: http://www.creoproject.com/. [Accessed: 03-Sep-2014].

[45] “Reduction of pollutant emissions from light vehicles.” [Online]. Available: http://europa.eu/legislation_summaries/environment/air_pollution/l28186_en.htm#amendingact. [Accessed: 01-Sep-2014].

[46] W. Piock, G. Hoffmann, A. Berndorfer, and P. Salemi, “Strategies Towards Meeting Future Particulate Matter Emission Requirements in Homogeneous Gasoline Direct Injection Engines,” *SAE Int. J. Engines*, vol. 4, no. 1, pp. 1455–1468, 2011.

[47] X. He, M. A. Ratcliff, and B. T. Zigler, “Effects of Gasoline Direct Injection Engine Operating Parameters on Particle Number Emissions,” *Energy & Fuels*, vol. 26, no. 4, pp. 2014–2027, Apr. 2012.

[48] M. Cary, “A model based engine calibration methodology for a port fuel injection, spark-ignition engine : the application of non-linear hierarchical statistical modelling and multi-objective optimisation techniques to the solution of real world multi-parameter engi,” PhD Thesis, University of Bradford, Bradford, 2003.

[49] M. A. Z. Khan, “Transient engine model for calibration using two-stage regression approach,” PhD Thesis, Loughborough University, 2011.

[50] K. Roepke, “Design of Experiments in Engine Development,” *Expert Verlag*, 2009.

[51] T. Kruse, S. Kurz, and T. Lang, “Modern Statistical Modeling and Evolutionary Optimization Methods for the Broad Use in ECU Calibration,” in *Advances in Automotive Control*, 2010, pp. 739–743.

[52] T. D. Barker, “ENGINE MAPPING TECHNIQUES,” *Int. J. Veh. Des.*, vol. 3, no. 2, pp. 142–152, 1982.

[53] Z. Waheed and N. Nadir, “Improvement of One Factor at a Time Through Design of Experiments,” in *Mathematical Applications in Engineering*, 2013, pp. 56–61.

[54] P. Klein, F. Kirschbaum, B. Hartmann, Y. Bogachik, and O. Nelles, “Adaptive Test Planning for the Calibration of Combustion Engines - Application,” in *Design of Experiments (DoE) in Engine Development*, 2013, pp. 17 – 30.

[55] J. Rango, T. Schnorbus, H. Kwee, R. Beck, B. Kinoo, S. Arthozoul, and M. Zhang, “Comparison of Different Approaches for Global Modeling of Combustion Engines,” in *Design of Experiments (DoE) in Engine Development*, 2013, pp. 70–91.

[56] B. Hartmann, W. Baumann, and O. Nelles, “Axes-Oblique Partitioning of Local Model Networks for Engine Calibration,” in *Design of Experiments (DoE) in Engine Development*, 2013, pp. 92 – 106.

[57] K. Roepke, A. Emtage, and D. Sampson, “Proposal for a Generic Interface between Test Bench Automation System and DoE Modelling Application,” *Des. Exp. Engine Dev. IV - Expert. - Libr. Tedesco - Libr. Univ.*, no. 9783816929376, 2009.

[58] I. Ryu, D. Han, S. Han, I. Hwang, and W. Kim, “Optimization of Catalyst Light-Off Time & Raw THC during Cold Start in Turbo DISI Engine Using DoE Method,” in *Design of Experiments in Engine Development*, 2007, pp. 133 – 144.

[59] L. A. D. Sheridan, R. Goyder, J. B. Cherrie, and T. M. Morton, “Defining a model-based calibration process for a twin-independent valve timing engine,” in *Proceedings of the IEEE International Conference on Control Applications*, 2004, vol. 2, pp. 1431–1436.

[60] A. C. Atkinson, A. N. Donev, and R. Tobias, *Optimum experimental designs, with SAS*. Oxford University Press, 2007.

[61] J. Seabrook, *Statistics for engine optimization*. Professional Engineering Pub., 2000, pp. 189–202.

[62] C. Atkinson and G. Mott, “Dynamic Model-Based Calibration Optimization: An Introduction and Application to Diesel Engines,” in *SAE World Congress*, 2005.

[63] A. Bolduan, D. Deifel, C. Eckstein, E. Kloppenburg, T. Huber, T. Kruse, and H. Ulmer, “From Local Towards Global: Models for Diesel Engine Calibration,” in *Design of Experiments (DoE) in Engine Development*, 2013, pp. 59–69.

[64] R. G. Regis, “Stochastic radial basis function algorithms for large-scale optimization involving expensive black-box objective and constraint functions,” *Comput. Oper. Res.*, vol. 38, no. 5, pp. 837–853, 2011.

[65] K. Crombecq, E. Laermans, and T. Dhaene, “Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling,” *Eur. J. Oper. Res.*, vol. 214, no. 3, pp. 683–696, Nov. 2011.

[66] D. Gorissen, K. Crombecq, W. Hendrickx, and T. Dhaene, “Adaptive Distributed Metamodeling,” in *High Performance Computing for Computational Science - VECPAR 2006*, Springer Berlin Heidelberg, 2007, pp. 579–588.

[67] F. Yates, “Sir Ronald Fisher and the Design of Experiments,” *Int. Biometric Soc.*, vol. 20, no. 2, pp. 307–321, 1964.

[68] G. Box and K. Wilson, “On the Experimental Attainment of Optimum Conditions,” *J. R. Stat. Soc. Ser. B*, vol. 13, no. 1, pp. 1–45, 1951.

[69] W. Charteris, “Taguchi’s system of experimental design and data analysis: a quality engineering technology for the food industry,” *Int. J. Dairy Technol.*, vol. 45, no. 2, pp. 33–49, May 1992.

[70] D. C. Montgomery, E. A. Peck, G. G. Vining, and J. Vining, “Introduction to linear regression analysis,” 2001.

[71] M. Guerrier and P. Cawsey, “The development of model based methodologies for gasoline IC engine calibration,” *SAE Trans.*, vol. 113, no. 3, pp. 981–1002, 2004.

[72] N. Didcock, A. Rainer, and S. Jakubek, *Optimization and Optimal Control in Automotive Systems*, vol. 455. Cham: Springer International Publishing, 2014, pp. 273–289.

[73] J. Seabrook, T. Salamon, S. Edwards, and I. Noell, “A comparison of Neural Networks, Stochastic Process Methods and Radial Basis Function for the Optimization of Engine Control Parameters,” in *Second Conference Design of Experiments in Engine Development*, 2003.

[74] D. M. Grove, D. C. Woods, and S. M. Lewis, “Multifactor B-spline mixed models in designed experiments for the engine mapping problem,” *J. Qual. Technol.*, vol. 36, no. 4, pp. 380–391, May 2004.

[75] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, “Design and Analysis of Computer Experiments,” *Stat. Sci.*, vol. 4, no. 4, pp. 409–423, Nov. 1989.

[76] E. Stinstra, D. den Hertog, P. Stehouwer, and A. Vestjens, “Constrained Maximin Designs for Computer Experiments,” *Technometrics*, vol. 45, no. 4, pp. 340–346, Nov. 2003.

[77] E. R. van Dam, B. Husslage, D. den Hertog, and H. Melissen, “Maximin Latin Hypercube Designs in Two Dimensions,” *Oper. Res.*, vol. 55, no. 1, pp. 158–169, Feb. 2007.

[78] K. Q. Ye, W. Li, and A. Sudjianto, “Algorithmic construction of optimal symmetric Latin hypercube designs,” *J. Stat. Plan. Inference*, vol. 90, no. 1, pp. 145–159, 2000.

[79] V. R. Joseph and Y. Hung, “ORTHOGONAL-MAXIMIN LATIN HYPERCUBE DESIGNS,” *Statistica Sinica 18*, 2008. [Online]. Available: http://www3.stat.sinica.edu.tw/statistica/oldpdf/A18n17.pdf.

[80] M. D. Morris and T. J. Mitchell, “Exploratory designs for computational experiments,” *J. Stat. Plan. Inference*, vol. 43, no. 3, pp. 381–402, 1995.

[81] M. E. Johnson, L. M. Moore, and D. Ylvisaker, “Minimax and maximin distance designs,” *J. Stat. Plan. Inference*, vol. 26, no. 2, pp. 131–148, 1990.

[82] B. Tang, “Orthogonal Array-Based Latin Hypercubes,” *J. Am. Stat. Assoc.*, vol. 88, no. 424, pp. 1392–1397, Dec. 1993.

[83] A. B. Owen, “Orthogonal arrays for computer experiments, integration and visualisation,” *Statistica Sinica 2*, 1992. [Online]. Available: http://www3.stat.sinica.edu.tw/statistica/oldpdf/a2n27.pdf.

[84] T. M. Cioppa and T. W. Lucas, “Efficient Nearly Orthogonal and Space-Filling Latin Hypercubes,” *Technometrics*, vol. 49, no. 1, pp. 45–55, Feb. 2007.

[85] “Introduction to Space-Filling Designs.” [Online]. Available: http://www.jmp.com/support/help/Introduction_to_Space-Filling_Designs.shtml. [Accessed: 26-Aug-2014].

[86] M. D. McKay, R. J. Beckman, and W. J. Conover, “A comparison of three methods for selecting values of input variables in the analysis of output from a computer code,” *Technometrics*, vol. 42, no. 1, pp. 55–61, 2000.

[87] T. W. Simpson, J. D. Poplinski, P. N. Koch, and J. K. Allen, “Metamodels for Computer-based Engineering Design: Survey and recommendations,” *Eng. Comput.*, vol. 17, no. 2, pp. 129–150, Jul. 2001.

[88] K. Fang, R. Li, and A. Sudjianto, *Design and modeling for computer experiments*. CRC Press, 2010.

[89] K. Crombecq, I. Couckuyt, D. Gorissen, and T. Dhaene, “Space-filling sequential design strategies for adaptive surrogate modelling,” in *The First International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering*, 2009.

[90] B. Steinberg and D. Lin, “A construction method for orthogonal Latin hypercube designs,” *Biometrika*, vol. 93, no. 2, pp. 279–288, 2006.

[91] F. A. C. Viana, G. Venter, and V. Balabanov, “An algorithm for fast optimal Latin hypercube design of experiments,” *Int. J. Numer. Methods Eng.*, vol. 82, no. 2, pp. 135–156, 2010.

[92] X. Liao, X. Yan, W. Xia, and B. Luo, “A fast optimal latin hypercube design for Gaussian process regression modeling,” in *Third International Workshop on Advanced Computational Intelligence*, 2010, pp. 474–479.

[93] H. Zhu, L. Liu, T. Long, and L. Peng, “A novel algorithm of maximin Latin hypercube design using successive local enumeration,” *Eng. Optim.*, vol. 44, no. 5, pp. 551–564, 2012.

[94] F. Fuerle and J. Sienz, “Formulation of the Audze–Eglais uniform Latin hypercube design of experiments for constrained design spaces,” *Adv. Eng. Softw.*, vol. 42, no. 9, pp. 680–689, Sep. 2011.

[95] A. Narayanan, V. V. Toropov, A. S. Wood, and I. F. Campean, “Simultaneous model building and validation with uniform designs of experiments,” *Eng. Optim.*, vol. 39, no. 5, pp. 497–512, Jul. 2007.

[96] D. Draguljić, T. J. Santner, and A. M. Dean, “Noncollapsing Space-Filling Designs for Bounded Nonrectangular Regions,” *Technometrics*, vol. 54, no. 2, pp. 169–178, May 2012.

[97] P. Audze and V. Eglais, “New approach for planning out of experiments,” *Probl. Dyn. Strengths*, vol. 35, pp. 104–107, 1977.

[98] W. C. R. Jin, R. Jin, W. Chen, and A. Sudjianto, “An efficient algorithm for constructing optimal design of computer experiments,” *J. Stat. Plan. Inference*, vol. 134, no. 1, pp. 268–287, 2005.

[99] S. C. Chuang and Y. C. Hung, “Uniform design over general input domains with applications to target region estimation in computer experiments,” *Comput. Stat. Data Anal.*, vol. 54, no. 1, pp. 219–232, 2010.

[100] T. D. K. Crombecq, L. D. Tommasi, D. Gorissen, “A Novel Sequential Design Strategy for Global Surrogate Modelling,” in *Simulation Conference*, 2009, pp. 731–742.

[101] B. Hartmann and O. Nelles, “Adaptive Test Planning for the Calibration of Combustion Engines - Methodology,” in *Design of Experiments (DoE) in Engine Development*, 2013, pp. 1 – 16.

[102] J.-S. Park, “Optimal Latin-hypercube designs for computer experiments,” *J. Stat. Plan. Inference*, vol. 39, no. 1, pp. 95–111, 1994.

[103] R. Lehmensiek, P. Meyer, and M. Muller, “Adaptive sampling applied to multivariate, multiple output rational interpolation models with application to microwave circuits,” *Int. J. RF Microw. Comput. Eng.*, vol. 12, no. 4, pp. 332–340, Jul. 2002.

[104] M. Sugiyama, “Active Learning in Approximately Linear Regression Based on Conditional Expectation of Generalization Error,” *J. Mach. Learn. Res.*, vol. 7, pp. 141–166, Dec. 2006.

[105] D. D. Geest, J. D., Dhaene, T., Fach, N., and Zutter, “Adaptive CAD-Model Building Algorithm for General Planar Microwave Structures,” *Microw. Theory Tech. IEEE Trans.*, vol. 47, no. 9, pp. 1801–1809, 1999.

[106] F. Provost, D. Jensen, and T. Oates, “Efficient progressive sampling,” in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’99*, 1999, pp. 23–32.

[107] L. Liu, “Could Enough Samples be more Important than Better Designs for Computer Experiments?,” in *38th Annual Simulation Symposium*, pp. 107–115.

[108] K. Crombecq, I. Couckuyt, D. Gorissen, and T. Dhaene, “Automated Response Surface Model Generation with Sequential Design Surrogate modelling,” pp. 1–15, 2009.

[109] K. Crombecq and T. Dhaene, “Generating Sequential Space-filling Designs Using Genetic Algorithms and Monte Carlo Methods,” in *Simulated Evolution and Learning*, Springer Berlin Heidelberg, 2010, pp. 80–84.

[110] D. Busby, C. L. Farmer, and A. Iske, “Hierarchical Nonlinear Approximation for Experimental Design and Statistical Data Fitting,” *SIAM J. Sci. Comput.*, vol. 29, no. 1, pp. 49–69, Jan. 2007.

[111] R. B. Gramacy and H. K. H. Lee, “Adaptive design an Analysis of supercomputer experiments,” *Technometrics*, vol. 51, no. 2, pp. 130–145, 2009.

[112] R. Jin, W. Chen, and A. Sudjianto, “ON SEQUENTIAL SAMPLING FOR GLOBAL METAMODELING IN ENGINEERING DESIGN,” *ASME 2002 Design Engineering Technical Conferences And Computers and Information in Engineering Conference*, 2002. [Online]. Available: http://ideal.mech.northwestern.edu/pdf/DAC_34092_chen.pdf.

[113] H. Niederreiter, “Random Number Generation and Monte Carlo Methods,” *Society for Industrial and Applied Mathematics*, 1992. [Online]. Available: http://www.springer.com/statistics/computational+statistics/book/978-0-387-00178-4.

[114] E. Rafajłowicz and R. Schwabe, “Halton and Hammersley sequences in multivariate nonparametric regression,” 2006.

[115] C. Q. Lam, “Sequential Adaptive Designs in Computer Experiments for Response Surface Model Fit,” PhD thesis Typescript, The Ohio State University, 2008.

[116] F. Aurenhammer, “Voronoi diagrams-a survey of a fundamental geometric data structure,” *ACM Comput. Surv.*, vol. 23, no. 3, pp. 345–405, Sep. 1991.

[117] P. Z. G. Qian, “Nested Latin hypercube designs,” *Biometrika*, vol. 96, no. 4, pp. 957–970, Sep. 2009.

[118] G. Rennen, B. Husslage, E. R. Van Dam, and D. Den Hertog, “Nested maximin Latin hypercube designs,” *Struct. Multidiscip. Optim.*, vol. 41, no. 3, pp. 371–395, Oct. 2009.

[119] F. Xiong, Y. Xiong, W. Chen, and S. Yang, “Optimizing Latin hypercube design for sequential sampling of computer experiments,” *Eng. Optim.*, vol. 41, no. 8, pp. 793–810, 2009.

[120] R. H. Myers, A. I. Khuri, and W. H. Carter, “Response Surface Methodology,” *Technometrics*, vol. 31, no. 2, pp. 137–157, May 1989.

[121] T. M. Morton and S. Knott, “Radial Basis Functions for Engine Modelling,” in *Statistics & Analytical Methods in Automotive Engineering, IMechE Paper C606/022/2002*, 2002.

[122] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006, pp. 79–128.

[123] M. T. Hagan and H. B. Demuth, “Neural networks for control,” in *Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)*, 1999, vol. 3, pp. 1642–1656.

[124] G. Matheron, *Advanced Geostatistics in the Mining Industry*. Dordrecht: Springer Netherlands, 1976, pp. 221–236.

[125] T. W. Simpson, T. M. Mauery, J. J. Korte, and F. Mistree, “Kriging Models for Global Approximation in Simulation-Based Multidisciplinary Design Optimization,” *AIAA J.*, vol. 39, no. 12, pp. 2233–2241, Dec. 2001.

[126] N. Cressie, “The origins of kriging,” *Math. Geol.*, vol. 22, no. 3, pp. 239–252, Apr. 1990.

[127] W. J. Welch, R. J. Buck, J. Sacks, H. P. Wynn, T. J. Mitchell, and M. D. Morris, “Screening, Predicting, and Computer Experiments,” *Technometrics*, vol. 34, no. 1, Mar. 1992.

[128] M. Meckesheimer, R. R. Barton, T. Simpson, F. Limayem, and B. Yannou, “Metamodeling of Combined Discrete/Continuous Responses,” *AIAA J.*, vol. 39, no. 10, pp. 1950–1959, Oct. 2001.

[129] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, Apr. 1980.

[130] S. Chen, E. S. Chng, and K. Alkadhimi, “Regularized orthogonal least squares algorithm for constructing radial basis function networks,” *Int. J. Control*, vol. 64, no. 5, pp. 829–837, Oct. 1996.

[131] Y. He and C. J. Rutland, “Application of artificial neural networks in engine modelling,” *Int. J. Engine Res.*, vol. 5, no. 4, pp. 281–296, Jan. 2004.

[132] R. J. Howlett, M. M. Zoysa, S. D. Walters, and P. A. Howson, “Neural Network techniques for monitoring and control of internal combustion engines’,” in *Int. Symposium on Intelligent Industrial Automation*, 1999.

[133] R. L. . Saunders, “Investigation to a Validation Methodology for Non-Parametric Models for Engine Mapping,” MSc thesis, University of Bradford, 2004.

[134] D. M. Grove, D. C. Woods, and S. M. Lewis, “Multifactor B-spline mixed models in designed experiments for the engine mapping problem,” *J. Qual. Technol.*, vol. 36, no. Compendex, pp. 380–391, 2004.

[135] G. G. Wang and S. Shan, “Review of Metamodeling Techniques in Support of Engineering Design Optimization,” *J. Mech. Des.*, vol. 129, no. 4, p. 370, Apr. 2007.

[136] A. Debavelaere, N. Sakushima, and M. Sidorkiewicz, “Multi-Objective Optimization of the Diesel Combustion System with Numerical DOEs,” in *Design of Experiments in Engine Development*, 2007, pp. 184 – 195.

[137] K. Deb, *Optimization for Engineering Design: Algorithms and Examples*. PHI Learning Pvt. Ltd., 2004.

[138] S. S. Rao, *Engineering optimization: theory and practice*. Wiley, 2009.

[139] A. Ravindran, K. M. Ragsdell, and G. V. Reklaitis, *Engineering optimization: Methods and applications*. Wiley-Interscience, 2006.

[140] K. Deb, *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing*. London: Springer London, 2011, pp. 3–34.

[141] J. Arora, *Introduction to Optimum Design*. Academic Press, 2004.

[142] M. A. Bhatti, *Practical optimization methods: with Mathematica applications*. Springer, 2000.

[143] P. T. Boggs and J. W. Tolle, “Sequential Quadratic Programming,” *Acta Numer.*, vol. 4, Nov. 2008.

[144] T. G. Kolda, R. M. Lewis, and V. Torczon, “Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods,” *SIAM Rev.*, vol. 45, no. 3, pp. 385–482, Jan. 2003.

[145] D. . Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, 1982.

[146] P. E. Gill and E. Wong, *Mixed Integer Nonlinear Programming*, vol. 154. New York, NY: Springer New York, 2012, pp. 147–224.

[147] P. Y. Papalambros and D. J. Wilde, *Principles of Optimal Design: Modeling and Computation*. Cambridge University Press, 2000, p. 390.

[148] R. Hooke and T. A. Jeeves, “`` Direct Search'' Solution of Numerical and Statistical Problems,” *J. ACM*, vol. 8, no. 2, pp. 212–229, Apr. 1961.

[149] P. K. Shukla and K. Deb, “On finding multiple Pareto-optimal solutions using classical and evolutionary generating methods,” *Eur. J. Oper. Res.*, vol. 181, no. 3, pp. 1630–1652, 2007.

[150] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[151] K. Deb and S. Agrawal, “Understanding interactions among genetic algorithm parameters,” *Foundations of Genetic Algorithms*. Morgan Kaufmann Publishers, San Mateo, Calif., pp. 265–286, 14-Feb-1999.

[152] K. Deb, “An introduction to genetic algorithms,” *Sadhana*, vol. 24, no. 4–5, pp. 293–315, Aug. 1999.

[153] K. Deb, *Multi-objective optimization using evolutionary algorithms*, Vol. 16. Chishester: John Wiley & Sons, 2001.

[154] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” in *IEEE International Conference on Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence*, 1998, pp. 69–73.

[155] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *International Conference on Neural Networks*, 1995, vol. 4, pp. 1942–1948.

[156] A. Narayanan, “Development of optimal Model-Building and Model-Validation Design of Experiments Using a Permutation Genetic algorithm,” MSc Thesis Typescript, University of Bradford, 2005.

[157] S. K. Iyer and B. Saxena, “Improved genetic algorithm for the permutation flowshop scheduling problem,” *Comput. Oper. Res.*, vol. 31, no. 4, pp. 593–606, 2004.

[158] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin: Springer Berlin Heidelberg, 1996.

[159] D. A. Coley, *An Introduction to Genetic Algorithms for Scientists and Engineers*. World Scientific, 1999, p. 227.

[160] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2003.

[161] K. Deb and R. B. Agrawal, “Simulated Binary Crossover for Continuous Search Space,” *Complex Syst.*, vol. 9, no. 2, pp. 115–148, 1995.

[162] J. Grefenstette, “Optimization of Control Parameters for Genetic Algorithms,” *IEEE Trans. Syst. Man. Cybern.*, vol. 16, no. 1, pp. 122–128, Jan. 1986.

[163] G. Rudolph, “Convergence analysis of canonical genetic algorithms.,” *IEEE Trans. Neural Netw.*, vol. 5, no. 1, pp. 96–101, Jan. 1994.

[164] J. Robinson, S. Sinton, and Y. Rahmat-Samii, “Particle swarm, genetic algorithm, and their hybrids: optimization of a profiled corrugated horn antenna,” in *IEEE Antennas and Propagation Society International Symposium (IEEE Cat. No.02CH37313)*, 2002, vol. 1, pp. 314–317.

[165] R. C. Eberhart and Y. Shi, “Particle swarm optimization: developments, applications and resources,” in *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, 2001, vol. 1, pp. 81–86.

[166] R. Poli, “Analysis of the Publications on the Applications of Particle Swarm Optimisation,” *J. Artif. Evol. Appl.*, no. 3, Jan. 2008.

[167] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II,” *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[168] N. Srinivas and K. Deb, “Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms,” *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, Sep. 1994.

[169] I. Das and J. E. Dennis, “A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems,” *Struct. Optim.*, vol. 14, no. 1, pp. 63–69, Aug. 1997.

[170] I. Das and J. E. Dennis, “Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems,” *SIAM J. Optim.*, vol. 8, no. 3, pp. 631–657, Aug. 1998.

[171] “Modeling the Engine Envelope - Model-Based Calibration Toolbox for MATLAB & Simulink - MathWorks United Kingdom.” [Online]. Available: http://www.mathworks.co.uk/products/mbc/description3.html.

[172] A. Messac and C. A. Mattson, “Normal Constraint Method with Guarantee of Even Representation of Complete Pareto Frontier,” *AIAA J.*, vol. 42, no. 10, pp. 2101–2111, May 2004.

[173] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[174] T. Murata and H. Ishibuchi, “MOGA: multi-objective genetic algorithms,” in *Proceedings of 1995 IEEE International Conference on Evolutionary Computation*, vol. 1, p. 289.

[175] P. CHANG, S. CHEN, and C. LIU, “Sub-population genetic algorithm with mining gene structures for multiobjective flowshop scheduling problems,” *Expert Syst. Appl.*, vol. 33, no. 3, pp. 762–771, Oct. 2007.

[176] E. Zitzler, K. Deb, and L. Thiele, “Comparison of multiobjective evolutionary algorithms: Empirical results,” *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.

[177] R. T. Haftka, J. H. Starnes Jr., F. W. Barton, and S. C. Dixon, “Comparison of Two Types of Structural Optimization Procedures for Flutter Requirements,” *AIAA J.*, vol. 13, no. 10, pp. 1333–1339, Oct. 1975.

[178] R. P. Henderson, J. R. R. A. Martins, and R. E. Perez, “Aircraft conceptual design for optimal environmental performance,” *Aeronaut. J.*, vol. 116, no. 1175, pp. 1–22, 2012.

[179] C. D. McAllister and T. W. Simpson, “Multidisciplinary Robust Design Optimization of an Internal Combustion Engine,” *J. Mech. Des.*, vol. 125, no. 1, pp. 124–130, Mar. 2003.

[180] M. Kokkolaras, L. Louca, G. Delagrammatikas, N. Michelena, Z. Filipi, P. Papalambros, J. Stein, and D. Assanis, “Simulation-based optimal design of heavy trucks by model-based decomposition: an extensive analytical target cascading case study - International Journal of Heavy Vehicle Systems - Volume 11, Numbers 3-4/2004 - Inderscience Publishers,” *Int. J. Heavy Veh. Syst.*, vol. 11, no. 3, pp. 403–433, 2004.

[181] I. Kroo, “MDO for Large-Scale Design,” in *Multidisciplinary Design Optimization: State of the Art*, SIAM, 1997, pp. 22–44.

[182] K. English, C. L. Bloebaum, and E. Miller, “Development of multiple cycle coupling suspension in the optimization of complex systems,” *Struct. Multidiscip. Optim.*, vol. 22, no. 4, pp. 268–283, Nov. 2001.

[183] J. Sobieszczanski-Sobieski, “Optimization by decomposition: A step from hierarchic to non-hierarchic systems,” *NASA STI/Recent Adv. Multidiscip. Anal. Optim.*, no. 1, pp. 51–78, Apr. 1989.

[184] J. Sobieszczanski-Sobieski, J. Agte, and J. R. Sandusky, “Bi-level integrated system synthesis (BLISS),” *Proc. 7th AIAA/USAF/NASA/ISSMO Symp. Multidiscip. Anal. Optim.*, no. 1, pp. 164–172, Jan. 1998.

[185] R. J. Balling and J. Sobieszczanski-Sobieski, “Optimization of coupled systems - A critical overview of approaches,” *AIAA J.*, vol. 34, no. 1, pp. 6–17, Jan. 1996.

[186] R. D. Braun, A. A. Moore, and I. M. Kroo, “Use of the Collaborative Optimization Architecture for Launch Vehicle Design,” Sep. 1996.

[187] J. Sobieszczanski-Sobieski, J. Agte, and R. Sandusky, “Bilevel Integrated System Synthesis,” *AIAA J.*, vol. 38, no. 1, pp. 164–172, Jan. 2000.

[188] N. Michelena, H. Park, and P. Y. Papalambros, “Convergence Properties of Analytical Target Cascading,” *AIAA J.*, vol. 41, no. 5, pp. 897–905, May 2003.

[189] H. M. Kim, D. K. D. Kumar, W. Chen, and P. Y. Papalambros, “Target Exploration for Disconnected Feasible Regions in Enterprise-Driven Multilevel Product Design,” *AIAA J.*, vol. 44, no. 1, pp. 67–77, Jan. 2006.

[190] B. A. Wujek, J. E. Renaud, S. M. Batill, and J. B. Brockman, “Concurrent Subspace Optimization Using Design Variable Sharing in a Distributed Computing Environment,” *Concurr. Eng.*, vol. 4, no. 4, pp. 361–377, Dec. 1996.

[191] R. S. Sellar, S. M. Batill, and J. E. Renaud, “Response surface based, concurrent subspace optimization for multidisciplinary system design,” in *Proceedings of the 34th AIAA Aerospace Sciences and Meeting Exhibit*, 1996.

[192] J. E. Renaud and G. A. Gabriele, “Approximation in nonhierarchic system optimization,” *AIAA J.*, vol. 32, no. 1, pp. 198–205, Jan. 1994.

[193] J. Shankar, R. T. Haftka, and L. T. Watson, “Computational study of a nonhierarchical decomposition algorithm,” *Comput. Optim. Appl.*, vol. 2, no. 3, pp. 273–293, Nov. 1993.

[194] N. P. Tedford and J. R. R. A. Martins, “Benchmarking multidisciplinary design optimization algorithms,” *Optim. Eng.*, vol. 11, no. 1, pp. 159–183, Mar. 2009.

[195] R. E. Perez, H. H. Liu, and K. Behdinan, “Evaluation of Multidisciplinary Optimization Approaches for Aircraft Conceptual Design (AIAA),” in *Proceedings of the 10th AIAA/ISSMO Muldisiciplinary Analysis and Optimization Conference*, 2004.

[196] S. I. Yi, J. K. Shin, and G. J. Park, “Comparison of MDO methods with mathematical examples,” *Struct. Multidiscip. Optim.*, vol. 35, no. 5, pp. 391–402, Jul. 2007.

[197] R. Braun, P. Gage, I. Kroo, and I. Sobiesk, “Implementation and Performance issues in Collaborative Optimization,” *Proc. 6th AIAA/USAF/NASA/ISSMO Multidiscip. Anal. Optim. Symp.*, vol. AIAA 1996-, 1996.

[198] A. DeMiguel and W. Murray, “An analysis of collaborative optimization methods,” in *proceeding of the 8th AIAA/USA/NASA/ISSMO symposium on Multidisciplinary Analysis and Optimization*, 2000.

[199] A. B. Cooper, P. Georgiopoulos, H. M. Kim, and P. Y. Papalambros, “Analytical Target Setting: An Enterprise Context in Optimal Product Design,” *J. Mech. Des.*, vol. 128, no. 1, 2006.

[200] N. M. Alexandrov and R. M. Lewis, “Analytical and Computational Aspects of Collaborative Optimization for Multidisciplinary Design,” *AIAA J.*, vol. 40, no. 2, pp. 301–309, May 2002.

[201] P. M. Zadeh, V. V. Toropov, and A. S. Wood, “Metamodel-based collaborative optimization framework,” *Struct. Multidiscip. Optim.*, vol. 38, no. 2, pp. 103–115, Aug. 2008.

[202] H. M. Kim, “Target Cascading in Optimal System Design,” PhD Thesis, University of Michigan, 2001.

[203] S. Tosserams, L. F. P. Etman, and J. E. Rooda, “Augmented Lagrangian coordination for distributed optimal design in MDO,” *Int. J. Numer. Methods Eng.*, vol. 73, no. 13, pp. 1885–1910, Mar. 2008.

[204] J. J. Michalek and P. Y. Papalambros, “An Efficient Weighting Update Method to Achieve Acceptable Consistency Deviation in Analytical Target Cascading,” *J. Mech. Des.*, vol. 127, no. 2, pp. 206–214, Mar. 2005.

[205] H. M. Kim, W. Chen, and M. M. Wiecek, “Lagrangian Coordination for Enhancing the Convergence of Analytical Target Cascading,” *AIAA J.*, vol. 44, no. 10, pp. 2197–2207, Oct. 2006.

[206] S. Tosserams, L. F. P. Etman, P. Y. Papalambros, and J. E. Rooda, “An augmented Lagrangian relaxation for analytical target cascading using the alternating direction method of multipliers,” *Struct. Multidiscip. Optim.*, vol. 31, no. 3, pp. 176–189, Feb. 2006.

[207] A. J. de Wit and F. Van Keulen, “Numerical Comparison of Multi-Level Optimization Techniques,” in *proceeding of the 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2007.

[208] C. Zhanjun, “Study on multidisciplinary hybrid hierarchic optimization method,” in *2011 IEEE 2nd International Conference on Computing, Control and Industrial Engineering*, 2011, vol. 1, pp. 286–289.

[209] X. Yin, F. Campean, A. Wood, W. Seale, and M. Goodman, “Automatic Optimisation Strategies for Diesel Engine Calibration,” in *in Design of Experiments in Engine Development, Expert Verlag, K Roepke (Editor)*, 2009, pp. 162 – 175.

[210] F. Campean, X. Yin, and A. Wood, “A Multi-Disciplinary Design optimisation Approach to Cam-Phasing Optimisation,” *CONAT 2010*, vol. 2, pp. 87–94, 2010.

[211] F. Millo, A. Perazzo, and E. Pautasso, “Optimizing the Calibration of a Turbocharged GDI Engine through Numerical Simulation and Direct Optimization,” *Int. J. Engines*, vol. 3, no. 1, pp. 556–570, 2010.

[212] D. Popovic, M. Jankovic, S. Magner, and A. R. Teel, “Extremum seeking methods for optimization of variable cam timing engine operation,” *IEEE Trans. Control Syst. Technol.*, vol. 14, no. Compendex, pp. 398–407, 2006.

[213] H. Kaji, K. Ikeda, and H. Kita, “Acceleration of parametric Multi-objective Optimization by an initialization technique for Multi-Objective Evolutionary Algorithms,” in *Evolutionary Computation, 2008. CEC 2008. (IEEE World*

Congress on Computational Intelligence). IEEE Congress on, 2008, pp. 2291–2297.

[214] O. Roudenko, C. Randazzo, L. Doria, and E. Castagna., “Application of a Pareto-Based evolutionary algorithm to fuel injection optimization,” in *International Conference on Statistics and Analytical Methods in Automotive Engineering*, 2002, vol. 4, pp. 81–92.

[215] Z. Lifeng, L. Yunqing, and M. Yunxia, “The application of multidisciplinary design optimization in IC engine’s energy saving,” in *International Technology and Innovation Conference (ITIC 2009)*, 2009.

[216] J. B. Heywood, “Internal combustion engine fundamentals,” 1988.

[217] D. F. Hagen and G. Holiday, “Effects of engine operating and design variables on exhaust emissions,” in *SAE*, 1976, pp. 206–223.

[218] T. Lancefield, R. Gayler, and A. Chattopadhay, “The Practical Application and Effects of a Variable Event Valve Timing System,” in *SAE on International Congress and Exposition*, 1993.

[219] F. Zhao, M.-C. Lai, and D. . Harrington, “Automotive spark-ignited direct-injection gasoline engines,” *Prog. Energy Combust. Sci.*, vol. 25, no. 5, pp. 437–562, Oct. 1999.

[220] D. M. D. M. Grove, D. C. D. C. Woods, and S. M. S. M. Lewis, “Multifactor B-spline mixed models in designed experiments for the engine mapping problem,” *J. Qual. Technol.*, vol. 36, no. 4, pp. 380–391, May 2004.

[221] T. P. Dwyer, “An Investigation into Improving the Repeatability of Steady-State Measurements From Nonlinear Systems,” PhD thesis, University of Bradford, 2015.

[222] P. Eastwood, *Particulate Emissions from Vehicles*. John Wiley & Sons, 2008.

[223] T. Beattie, R. Osborne, and W. Graupner, “Engine Test Data Quality Requirements for Model Based Calibration: A Testing and Development Efficiency Opportunity,” in *SAE Technical Paper*, 2013, pp. 10.4271/2013–01–0351.

[224] R. Kozan and M. Gokce, “Two-Stage Engine Mapping for the Calibration of Carbon Monoxide Emission,” *Modern Applied Science*, vol. 3, no. 4. p. P30, 2009.

[225] T. G. Leone, E. J. Christenson, and R. A. Stein, “Comparison of Variable Camshaft Timing Strategies at Part Load,” SAE International, SAE Technical Paper 960584, Feb. 1996.

[226] “AVL CAMEO Model and Map - avl.com.” [Online]. Available: https://www.avl.com/cameo-model-and-map. [Accessed: 25-Apr-2014].

[227] A. M. Andrew, “Another efficient algorithm for convex hulls in two dimensions,” *Inf. Process. Lett.*, vol. 9, no. 5, pp. 216–219, Dec. 1979.

[228] “Global Optimization Toolbox - MATLAB - MathWorks United Kingdom.” [Online]. Available: http://www.mathworks.co.uk/products/global-optimization/. [Accessed: 31-Aug-2014].

[229] L. Wang, S. Shan, and G. G. Wang, “Mode-pursuing sampling method for global optimization on expensive black-box functions,” *Eng. Optim.*, vol. 36, no. 4, pp. 419–438, Aug. 2004.

[230] B. G. M. Husslage, G. Rennen, E. R. Dam, and D. Hertog, “Space-filling Latin hypercube designs for computer experiments,” *Optim. Eng.*, vol. 12, no. 4, pp. 611–630, Nov. 2010.

[231] A. B. Lambe and J. R. R. A. Martins, “Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes,” *Struct. Multidiscip. Optim.*, vol. 46, no. 2, pp. 273–284, Jan. 2012.

[232] T. R. Browning, “Applying the design structure matrix to system decomposition and integration problems: a review and new directions,” *IEEE Trans. Eng. Manag.*, vol. 48, no. 3, pp. 292–306, 2001.

[233] M. G. Sahab, “Cost Optimization of Reinforced Concrete Flat Slab Buildings,” PhD Thesis Typescript, University of Bradford, 2002.

[234] Z. Michalewicz and M. Schoenauer, “Evolutionary Algorithms for Constrained Parameter Optimization Problems,” *Evol. Comput.*, vol. 4, no. 1, pp. 1–32, Mar. 1996.

[235] E. Mezura-Montes and C. A. Coello Coello, “Constraint-handling in nature-inspired numerical optimization: Past, present and future,” *Swarm Evol. Comput.*, vol. 1, no. 4, pp. 173–194, Dec. 2011.

[236] G. E. LIEPINS and M. D. VOSE, “Representational issues in genetic optimization,” *J. Exp. Theor. Artif. Intell.*, vol. 2, no. 2, pp. 101–115, Apr. 1990.

[237] H. . Schwefel, *Evolution and Optimum Seeking: The Sixth Generation*. New York, NY, United States: John Wiley & Sons, Inc., 1993.

[238] O. Yeniay, “Penalty function methods for constrained optimization with genetic algorithms,” *Math. Comput. Appl.*, vol. 10, no. 1, pp. 45–56, 2005.

[239] H. Fang and F. H. Mark, “Metamodeling with radial basis functions,” in *46th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference*, 2005, pp. 2005–2059.

[240] “Lessons from Minitab 17 Help - Minitab.” [Online]. Available: http://www.minitab.com/en-us/academic/teaching-resources/lessons-from-minitab-help/. [Accessed: 16-Nov-2014].

[241] H. Stuhler, T. Kruse, A. Stuber, K. Gschweitl, W. Piock, H. Pfluegl, and P. Lick, “Automated Model-Based GDI Engine Calibration Adaptive Online DOE Approach,” *SAE Pap.*, no. 2002–01–0708, pp. 2002–01–0708, 2002.

[242] H. J. Rajan, J. Kelly, H. Hoetzendorfer, N. Keuth, H. Pfluegl, T. Winsel, and S. Roeack, “Industrialization of base calibration methods for ECU-functions exemplary for air charge determination,” in *SAE Paper*, 2010, pp. 2010–01–0331.

[243] M. R. Kianifar, “CREO-12032012: Preliminary Report on Phase 1 Screening DoE Results,” University of Bradford, 2012.

[244] “RESIDUAL ANALYSIS IN MULTIPLE REGRESSION.” [Online]. Available: http://canmedia.mcgrawhill.ca/college/olcsupport/bowerman/2ce/OptionalContent/bow02371_OLC_12_13.pdf.

[245] M. R. Kianifar, “CREO-15032013: Preliminary Report on Phase 2 Screening DoE Results for Multiple Injections,” University of Bradford, 2013.

[246] C. J. Brace, S. Akehurst, and M. C. Ward, “The use of prior knowledge to accelerate the determination of the permissible operating envelope of an internal combustion engine,” *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.*, vol. 225, no. 2, pp. 206–221, Feb. 2011.

[247] “NGPM -- A NSGA-II Program in Matlab v1.4 - File Exchange - MATLAB Central.” [Online]. Available: http://www.mathworks.co.uk/matlabcentral/fileexchange/31166-ngpm-a-nsga-ii-program-in-matlab-v1-4. [Accessed: 23-Sep-2014].

[248] MathWorks, “fmincon - Matlab R2013b,” 2014. [Online]. Available: http://www.mathworks.com.

[249] MathWorks, “Global Optimization Toolbox - ga - Matlab R2013b.” [Online]. Available: http://www.mathworks.com.

[250] S. Tosserams, M. Kokkolaras, L. F. P. Etman, and J. E. Rooda, “A Nonhierarchical Formulation of Analytical Target Cascading,” *J. Mech. Des.*, vol. 132, no. 5, pp. 1–13, 2010.

[251] N. Kang, M. Kokkolaras, and P. Papalambros, “Optimal Design of Commercial Vehicle Systems Using Analytical Target Cascading,” *12th AIAA Aviat. Technol. Integr. Oper. Conf. 14th AIAA/ISSMO Multidiscip. Anal. Optim. Conf.*, pp. 10.2514/6.2012–5524, Sep. 2012.

[252] N. Kang, M. Kokkolaras, and P. Y. Papalambros, “Solving multiobjective optimization problems using quasi-separable MDO formulations and analytical target cascading,” in *10th World Congress on Structural and Multidisciplinary Optimization*, 2013.

[253] M. R. Kianifar, L. F. Campean, and D. Richardson, “Sequential DoE Framework for Steady State Model Based Calibration,” *SAE Int. J. Engines*, vol. 6, no. 2, pp. 843–855, Apr. 2013.

[254] R. J. Lygoe, “Complexity Reduction in High-Dimensional Multi-Objective Optimisation,” PhD thesis, University of Sheffield, 2010.

[255] J.-P. Li and A. S. Wood, “An adaptive species conservation genetic algorithm for multimodal optimization,” *Int. J. Numer. Methods Eng.*, vol. 79, no. 13, pp. 1633–1661, Sep. 2009.

List of Publications

- **Journals:**

(1) M. R. Kianifar, I. F. Campean, and D. Richardson, “Sequential DoE Framework for Steady State Model Based Calibration,” *SAE Int. J. Engines*, vol. 6, no. 2, pp. 843–855, Apr. 2013.

(2) M. R. Kianifar and I. F. Campean, “Analytical Target Cascading for Engine Calibration Optimisation,” *Int. J. Powertrains*, vol. 3, no. 3, pp. 279–302, 2014.

- **Chapter of Book**

(1) T. P. Dwyer, M. R. Kianifar, W. J. Bradley, I. F. Campean, B. A. Mason, M. K. Ebrahimi, D. Richardson, and L. Beddow, “DoE Framework for GDI Engine Mapping and Calibration Optimisation for CO 2 and Particulate Number Emissions,” *Roepke Des. Exp. Engine Dev. Expert Verlag*, pp. 418–432, 2013.

- **Reviewed Conference Papers**

(1) M. R. Kianifar and I. F. Campean, “APPLICATION OF ANALYTICAL TARGET CASCADING FOR ENGINE CALIBRATION OPTIMIZATION PROBLEM,” in *ASME Proceedings of the 40th Design Automation Conference*, 2014.

(2) M. R. Kianifar, I. F. Campean, T. Beattie, and D. Richardson, "Analytical Target Cascading Framework for Diesel Engine Calibration Optimisation," SAE Technical Paper 2014-01-2583, 2014.

(3) M. R. Kianifar, I. F. Campean, and D. Richardson, "Evaluation of Camshaft Control Strategies for a GDI Engine using a Multidisciplinary Optimisation Framework," SAE Technical Paper 2014-01-2581, 2014.

- **Conferences**

(1) M. R. Kianifar and I. F. Campean, “A Sequential Model-Building Model-Validation DoE for Steady State Model Based Calibration.” Bradford, United Kingdom, 2012.

(2) M. R. Kianifar and I. F. Campean, “Multi-Disciplinary Optimisation for Variable Camshaft Timing Engine Calibration Optimisation.” Bradford, United Kingdom, 2012.

Appendices

Appendix I

An example of the EXCEL data sheet, containing the steady state data collected from University of Bradford dynamometer for the GDI engine with single injection strategy:

Date	DoE Test	Ignition (CA BTDC)	(CA ATDC)	EVC (CA ATDC)	FRP (MPa)	SoI (CA BTDC)	Pn (p/ccm)	THC (ppm)	CO (vol%)	NOx (vol%)	Ex_temp (deg C)	FB_VAL (kg/hr)	IMEP_eng (bar)	IMEP_sd (bar)	IMEP_cv
120516	1	35.25	13.5	4	11.5686	332.05882	1008332	0.218905	0.75659	0.259655	534.5886	6.1924	2.515225	0.02137	0.0085
120516	2	41.25	-4.5	10.5	11.7059	264.4	1395304	0.201514	0.90653	0.167539	537.2111	6.1477	2.551807	0.03441	0.01349
120516	3	51.75	22	42	14.3137	298.23529	782244.9	0.217346	0.72436	0.097317	528.1773	5.9221	2.556209	0.03268	0.01278
120516	4	41.25	17	27.5	13.3529	273.23	1076266	0.208597	0.63021	0.161779	538.3243	6.0071	2.526487	0.0225	0.00891
120516	5	32.25	16	-3	13.902	280.58824	1008630	0.215127	0.62599	0.271222	528.5067	6.229	2.528305	0.01341	0.00531
120516	6	36.75	-6	1	12.6667	310	985141.3	0.21069	0.63197	0.158203	535.2454	6.1624	2.535464	0.02272	0.00896
120516	7	35.25	24.5	9.5	8.13725	292.36	1764370	0.227206	0.65352	0.248953	533.5	6.1465	2.518778	0.01768	0.00702
120516	8	33.75	27	-2	13.0784	317.34	970114.1	0.213375	0.6083	0.2699	528.6022	6.1708	2.500867	0.02141	0.00856
120516	9	46.5	36.5	37	10.8824	287.93	1351853	0.20462	0.60577	0.130011	539.3056	5.8997	2.538323	0.02005	0.0079
120516	10	45	6	25.5	13.4902	308.52941	956748.4	0.197517	0.6069	0.141917	540.4904	6.0294	2.553413	0.023	0.00901
120516	11	36.75	45	19.5	8.82353	299.72	1599284	0.210787	0.73463	0.207559	534.3478	6.0676	2.517074	0.02491	0.0099
120516	12	36	29.5	15.5	14.0392	283.52	1157786	0.212951	0.63311	0.226208	534.2351	6.0821	2.506286	0.02293	0.00915
120516	13	33.75	42.5	7.5	14.8627	295.29412	982098.1	0.209931	0.66227	0.250647	532.9798	6.1419	2.503119	0.01602	0.0064
120516	14	34.5	8.5	0	9.09804	271.76471	1724624	0.207374	0.8274	0.250649	529.1747	6.2285	2.50965	0.01783	0.0071
120516	15	34.5	21	6	9.64706	293.81	1614772	0.215615	0.78551	0.247852	530.815	6.2152	2.534596	0.0132	0.00521
120424	16	36.75	10	12.5	13.7647	327.66	1014891	0.206202	0.60511	0.223919	544.1397	6.1087	2.55386	0.02376	0.0093
120424	17	42	2.5	17.5	10.3333	333.54	1755097	0.205686	0.80433	0.170409	540.8466	6.0654	2.564321	0.02441	0.00952
120424	18	66.75	-2.5	38	10.1961	289.41176	1499633	0.270657	0.74978	0.054774	529.8603	5.949	2.614888	0.0584	0.02233
120424	19	34.5	35.5	11.5	8.27451	320.3	1678698	0.21743	0.64432	0.239543	537.6052	6.0513	2.528876	0.02039	0.00806
120424	20	33.75	12.5	-4	8.96078	307.05	1581977	0.221555	0.64554	0.267529	531.6427	6.169	2.520519	0.02378	0.00943
120424	21	58.5	11	43	11.8431	312.94118	979898	0.229827	0.59405	0.069341	530.626	5.8687	2.5885	0.03812	0.01473
120424	22	37.5	-7	2	10.4706	302.64706	1420924	0.219232	0.73043	0.207423	535.7902	6.126	2.56001	0.02508	0.0098
120424	23	45	-1	20.5	9.78431	261.47059	1735753	0.199981	0.9594	0.126128	538.6614	6.1129	2.587576	0.04106	0.01587
120424	25	41.25	49	33	10.7451	279.11765	1453664	0.209861	0.64787	0.136037	540.348	5.8829	2.543928	0.02767	0.01088
120424	26	38.25	46.5	26.5	12.8039	268.83	1271611	0.211826	0.54534	0.176892	541.5213	5.9186	2.533537	0.01546	0.0061
120424	27	44.25	15	29.5	11.9804	330.58824	1067847	0.200423	0.63383	0.144665	543.3254	5.9906	2.57444	0.02216	0.00861
120424	28	33.75	41.5	8.5	10.6078	321.75	1239912	0.213039	0.73229	0.239201	538.6667	6.0884	2.521974	0.02767	0.01097
120424	29	45.75	-8.5	18.5	14.5882	286.48	1015994	0.216581	0.72723	0.099988	538.857	5.9934	2.59181	0.0275	0.01061
120424	30	41.25	38	30.5	11.4314	324.7	1178452	0.20286	0.61205	0.146143	541.9589	5.8832	2.528374	0.02107	0.00833
120424	31	34.5	34	13.5	13.2157	314.41176	984645.8	0.211509	0.55753	0.218429	541.7246	6.0781	2.543884	0.01759	0.00691
120424	32	44.25	28	36	9.5098	285	1592965	0.211445	0.69793	0.120324	542.2384	5.9432	2.570319	0.02824	0.01099
120424	33	33.75	5	3	11.0196	270.28	1464834	0.209855	0.8329	0.22578	533.4695	6.1839	2.538102	0.0285	0.01123
120424	34	34.5	32	14.5	10.0588	267.35294	1509482	0.215806	0.75919	0.218217	531.2031	6.02	2.503929	0.02482	0.00991
120424	35	32.25	40.5	-5	12.5294	296.77	1137929	0.216074	0.63715	0.266909	531.078	6.1653	2.528261	0.01844	0.00729
120424	36	48.75	-3.5	23.5	14.1765	326.17647	881770.8	0.202032	0.6911	0.111499	542.0478	5.9829	2.587289	0.03027	0.0117
120424	37	42.75	39	34	14.7255	323.25	687326.8	0.198619	0.71623	0.141983	541.4701	5.9055	2.550628	0.0223	0.00874
120424	38	47.25	25.5	41	12.2549	277.64	1181420	0.212856	0.52693	0.104735	539.0906	5.8698	2.570218	0.02849	0.01108
120424	39	36.75	18.5	16.5	11.2941	301.17	1232191	0.209733	0.71409	0.211603	545.3186	6.0979	2.552522	0.03291	0.01289
120424	40	33	1.5	-1	12.3922	276.19	1204526	0.209183	0.72971	0.245528	531.4382	6.1473	2.5226	0.01536	0.00609
120425	41	54.75	0	31.5	11.1569	305.6	1170883	0.225695	0.71022	0.081887	534.5088	5.9165	2.573496	0.03673	0.01427
120425	42	52.5	4	32	8.68627	265.88	1795537	0.229805	0.77097	0.095361	527.1059	5.9277	2.557397	0.04132	0.01616
120425	43	42.75	7.5	22.5	8.41176	329.11	1692556	0.213915	0.70381	0.159674	537.0151	5.9449	2.54522	0.02215	0.0087
120425	44	44.25	44	39	13.6275	304.13	872487.5	0.210882	0.57541	0.114747	537.9158	5.7935	2.545633	0.02534	0.00996
120425	45	39	33	28.5	8.54902	327.571	1752866	0.217741	0.68622	0.158391	536.6921	5.8875	2.52764	0.02197	0.00869
120425	46	33	47.5	5	12.1176	290.88	1223933	0.225051	0.67866	0.255788	531.4817	6.1161	2.51384	0.0156	0.0062
120425	47	32.25	30.5	7	14.451	262.95	1117732	0.206019	0.72518	0.237109	534.3571	6.095	2.533249	0.01884	0.00744
120425	48	38.25	23.5	21.5	9.92157	318.82	1449216	0.216504	0.72989	0.196127	539.366	6.01	2.545128	0.02502	0.00983
120425	49	51	19.5	40	9.23529	311.46	1476816	0.219867	0.69052	0.09626	534.9124	5.8416	2.55876	0.02954	0.01154
120425	50	54.75	-9.5	24.5	12.9412	282.07	1212799	0.235746	0.60955	0.083336	532.1998	5.8866	2.581882	0.0398	0.01541

Appendix II

Local calibration candidates at each minimap point:

Candidate	IVO	EVC	FRP	SOI	FC (kg/hr)	Pn(pccm)
1	35.2	14.8	12.3	294	1.374	524
2	35.2	14.8	12.3	294	1.374	400
3	35.4	15.4	12.2	295	1.374	284
4	35.3	15.1	12.2	295	1.374	135

Local candidates for 700-28 operating point with single injection

Candidate	IVO	EVC	FRP	SOI	FC (kg/hr)	Pn(pccm)
1	30.4	24.6	11.1	311	3.192	6594
2	29.2	25	11.4	310	3.2	3445
3	29.5	24.95	11.7	311	3.207	1664
4	34.7	23.6	11.9	314	3.212	358

Local candidates for 1500-41 operating point with single injection

Candidate	IVO	EVC	FRP	SOI	FC (kg/hr)	Pn(pccm)
1	3.2	40.3	12.5	307	4.66	5.45×10^5
2	11.6	43.3	12.7	309	4.666	4.97×10^5
3	14.7	43.3	13	307	4.674	4.61×10^5
4	19.3	42.8	13.5	306	4.685	4.27×10^5

Local candidates for 1250-125 operating point with single injection

Candidate	IVO	EVC	FRP	SOI	FC (kg/hr)	Pn(pccm)
1	-9.5	33.2	10.6	296	5.027	1.00×10^6
2	-4.7	30.3	13.9	281	5.103	7.81×10^5
3	29.6	37	14.1	318	5.154	6.58×10^5
4	27.5	30.5	14.6	318	5.192	5.89×10^5

Local candidates for 1500-105 operating point with single injection

Candidate	IVO	EVC	FRP	SOI	FC (kg/hr)	Pn(pccm)
1	-3.5	-3.4	9	299	5.515	1.53×10^6
2	25.4	38.5	14.7	306	5.848	5.67×10^5
3	35.1	19.3	14.9	324	6.044	5.60×10^5
4	33.9	8.7	14.7	313	6.174	4.73×10^5

Local candidates for 2000-81 operating point with single injection

Candidate	IVO	EVC	FRP	SOI	FC (kg/hr)	Pn(pccm)
1	37	41.1	12.4	288	10.69	1.60×10^6
2	37.1	41.1	12.8	293	10.71	1.50×10^6
3	35.7	40.6	13.2	300	10.73	1.40×10^6
4	33.8	37.4	13.6	306	10.76	1.29×10^6

Local candidates for 2000-199 operating point with single injection

Candidate	IVO	EVC	FRP	SOI1	Delay	Split	FC (kg/hr)	Pn(pccm)
1	-4.7	39.8	13	318	4.42	0.5	4.556	7.7×10^5
2	-1.3	40.7	13.4	320	4.1	0.51	4.582	7.2×10^5
3	5	40.2	13.7	329	3.2	0.52	4.624	6.48×10^5
4	12.5	39.4	14	337	3.07	0.54	4.668	6.03×10^5

Local candidates for 1250-125 operating point with double injections

Candidate	IVO	EVC	FRP	SOI1	Delay	Split	FC (kg/hr)	Pn(pccm)
1	6.4	34.5	12.9	345	7.03	0.57	5.036	1.11×10^6
2	10.1	36.1	13.2	335	7.62	0.56	5.055	1.03×10^6
3	49.9	35.4	13.3	312	2.07	0.55	5.083	8.82×10^5
4	42.7	34.4	13.4	324	1.58	0.59	5.113	7.82×10^5

Local candidates for 1500-105 operating point with double injections

Candidate	IVO	EVC	FRP	SOI1	Delay	Split	FC (kg/hr)	Pn(pccm)
1	44.8	42.5	13.3	310	3.48	0.6	5.87	1.78×10^6
2	35.4	43.1	13.1	327	2.55	0.59	5.902	1.54×10^6
3	23.3	41	13	341	2.23	0.58	5.937	1.33×10^6
4	23.2	38.3	13.1	347	1.57	0.54	5.96	1.19×10^6

Local candidates for 2000-81 operating point with double injections

Candidate	IVO	EVC	FRP	SOI1	Delay	Split	FC (kg/hr)	Pn(pccm)
1	11.2	43.3	13.1	338	3.16	5.52	10.65	2.08×10^6
2	18	42.6	13.3	344	2.89	0.52	10.66	1.93×10^6
3	16.6	40.4	13.3	350	3.01	0.55	10.68	1.82×10^6
4	23.1	36.9	13.4	354	2.28	0.56	10.73	1.64×10^6

Local candidates for 2000-199 operating point with double injections